AN ANALOGUE FRONT-END TEST-BED FOR SOFTWARE DEFINED RADIO

Vincent J. Arkesteijn, Eric A.M. Klumperink and Bram Nauta

MESA$^+$ Research Institute, IC-Design Group
University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands
v.j.arkesteijn@utwente.nl, http://icd.el.utwente.nl/

A Software Defined Radio (SDR) is a radio receiver and/or transmitter, whose characteristics can to a large extent be defined by software. Thus, an SDR can receive and/or transmit a wide variety of signals, supporting many different standards.

In our research, we currently focus on a demonstrator that is able to receive both Bluetooth and HiperLAN/2. This helps us to identify problems associated with SDR, and will provide a test-bed for possible solutions to these problems. The two standards differ significantly in characteristics like frequency band, signal bandwidth and modulation type. Combining two different standards in one receiver appears to pose new design challenges. For example, in the wide frequency range that we want to receive, many strong signals may exist. This leads to severe linearity requirements for wideband receivers.

This paper describes some receiver architectures. One design has been selected. This receiver has been built, and some measurement results are included.

1 INTRODUCTION

A Software Radio is a radio receiver and/or transmitter implemented fully in software. Because software runs on digital hardware and radio waves are analogue by nature, an analogue-to-digital converter is usually included. Due to technology constraints however, this approach is infeasible.

In recent years, interest for Software *Defined* Radio (SDR) has been increasing, as indicated for example by [2]. In a Software Defined Radio, all relevant functions of the radio can be defined (controlled, programmed) by software. This does not however necessarily mean that all functions are implemented in software, as in a Software Radio.

Software Defined Radio can bring many advantages. One advantage is the convenience for the user. Having a multi-standard terminal (mobile telephone, laptop with wireless LAN interface) enables global roaming, without carrying an abundance of hardware.

A second advantage is a shorter development time and cost for the manufacturer[3]. Assuming that software can be developed faster than hardware, a Software Defined

	Bluetooth[6]	HiperLAN/2[5]
band	2.4 – 2.48 GHz	5.15 – 5.725 GHz
ch. bandwidth	$\sim$ 600 kHz	$\sim$ 16 MHz
ch. spacing	1 MHz	20 MHz
nom. bitrate	1 Mb/s	6 – 54 Mb/s
modulation	GFSK	QAM+OFDM
mult. access	FHSS	TDMA
duplex	TDD	TDD

Table 1: Some characteristics of Bluetooth and HiperLAN/2

Radio can be upgraded to a new standard, a new version of the standard or fitted with a better filter much faster than a conventional radio.

A last advantage of Software Defined Radio mentioned here, is its adaptability to a dynamic environment[4]. A Software Defined Radio can dynamically make a trade-off between performance and energy consumption. By minimizing the performance (while still maintaining a required quality of service), battery life can be maximized.

In our project[1], we aim at SDR front-end hardware. Two groups are involved; the IC-Design group concentrates on the analogue part of the front-end, the Laboratory Signals and Systems on the digital part. This paper focusses on the analogue part.

In order to locate typical SDR-related problems, and to have a test-bed for possible solutions to these problems, it was decided to build a demonstrator. This demonstrator shall be capable of receiving Bluetooth[6] and HiperLAN/2[5] signals and of demodulating them correctly. Some characteristics of these two standards are shown in table 1. As can be seen, these standards differ considerably, which should help in identifying typical SDR-problems.

In the next section, three architectures are presented, and one is selected. The selected design has been built, and section 3 discusses some results. Finally, conclusions are drawn and some ideas for further research are presented.

2 ARCHITECTURE CONSIDERATIONS

This section describes some of the design challenges in designing a Software Defined Radio. This is done by starting with a very simple and flexible receiver, and gradually changing this into an architecture that is feasible with current technology.

The first architecture to be considered is an ideal software radio. This is shown in figure 1. The antenna signal is filtered, amplified by the low noise amplifier (LNA) and

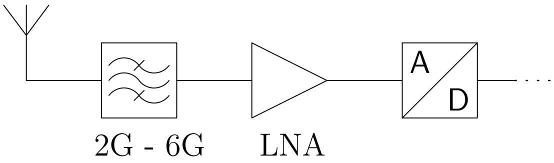


Figure 1: A Software Radio front-end

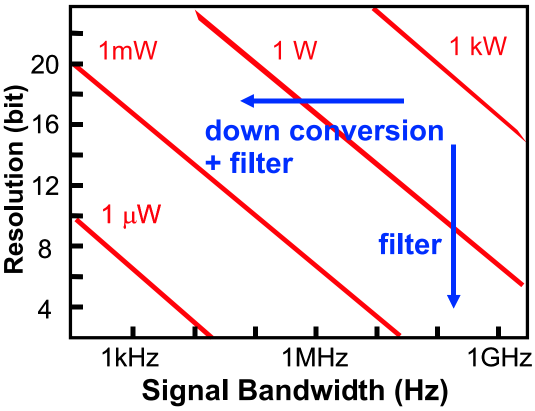


Figure 2: Power consumptions of ADC's as a function of signal bandwidth and resolution

converted to digital by the analogue-to-digital converter (ADC). As the bandwidth is 4 GHz, this would require an ADC with a sample rate of at least 8 GHz.

Furthermore, the required resolution would be very high, as can be seen as follows. Signals of up to 0 dBm may be present at the receiver input[5]. At the same time, the maximum input noise to the demodulator is around −164 dBm/Hz, or −68 dBm/4 GHz. This requires an SNR of 68 dB, corresponding to 12 bits of resolution.

In [7], a statistical analysis shows a strong dependancy of power concumption on sample rate and resolution of ADC's. This gives rise to a figure of merit

$$\mathrm{FoM} = \frac{2^{\mathrm{SNRbits}} \mathrm{f}_{\mathrm{sample}}}{\mathrm{P}}$$

where SNRbits is the resolution, $\mathrm{f}_{\mathrm{sample}}$ the sample rate in Hz and P the dissipated power in W.

Using a (conservative) FoM of 10 pJ/conversion results in figure 2. This figure shows that the above mentioned combination of sample rate and resolution (12 bits @ 8 GHz) would lead to excessive power consumption, if it were feasible at all. This is not expected to change significantly in the near future[7].

To relax the requirement on the ADC, a second architecture is proposed. This is

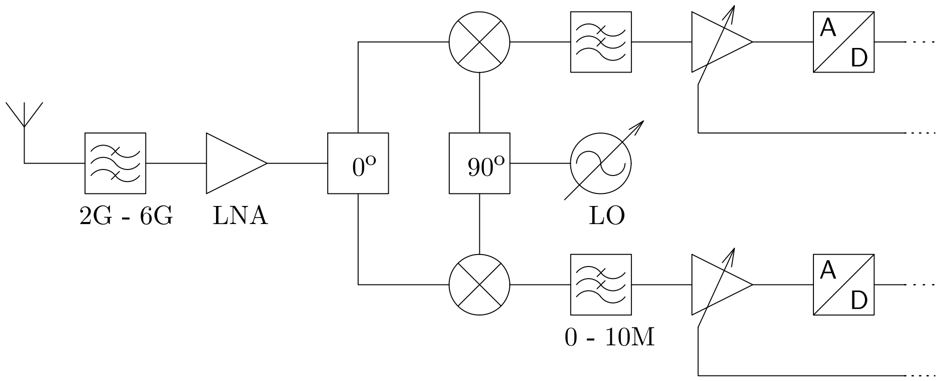


Figure 3: A Software Defined Radio front-end with one wide RF filter

shown in figure 3. Since the ADC is preceded by a downconverter and a low-pass filter, sample rate and resolution requirements are relaxed.

A problem still remains, however. Both the Bluetooth and the HiperLAN/2-standard specify out-of-band signal levels at which compliant receivers have to maintain a certain bit error rate. These levels are such that in a single-band receiver, these signals can be attenuated by a simple second or fourth order bandpass filter, and therefore do not present a problem. In this receiver however, these out-of-band signals are not attenuated. This results in extremely high linearity requirements. It was calculated for instance, that an IIP_2 of +82 dBm and an IIP_3 of +36 dBm were required. This was deemed unfeasible, based on a literature study of stade-of-the-art integrated front-ends.

To relax linearity requirements on the LNA and mixer, a third architecture is presented. See figure 4. Instead of one RF filter, two are now present. These filters attenuate strong out-of-band unwanted signals. This leads to feasible linearity requirements.

Of course, this limits the flexibility of the architecture. But since one antenna covering the whole frequency range would also be problematical, especially when also transmitting, a switch would be required anyway. An option would be to integrate everything on one chip, excluding the antennas and filters. This way, development of a receiver for a new standard would still be sped up, because only the antenna and filter would have to be designed.

The presented architecture is a low-IF receiver when used for Bluetooth reception, and a zero-IF receiver when used for HiperLAN/2 reception.

3 IMPLEMENTATION

The architecture presented in figure 4 has largely been built. The antennas, RF filters and band switch have been omitted. The rest of the receiver (LNA, power split-

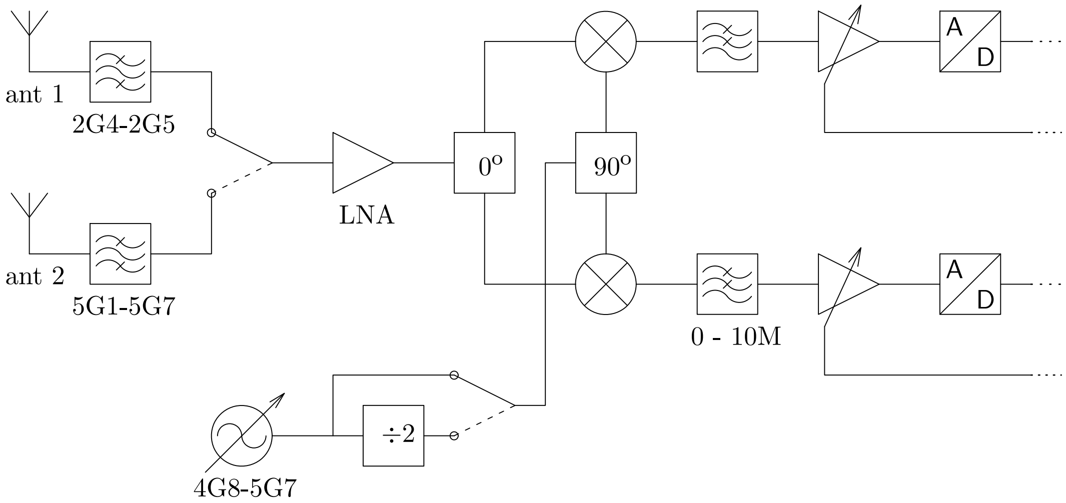


Figure 4: A Software Defined Radio front-end with switchable RF filters. The two switches, the LO frequency and the gain of the IF amplifiers are software defined.

ters, mixers, filters) has been built. All components are on separate boards, connected together using coaxial connectors. This facilitates easy experimentation with different architectures. The following components have been used.

LNA	Mini-Circuits ERA-2
power splitter	Mini-Circuits ZN2PD-9G
mixers	Mini-Circuits MBA-671
90^o power splitter	Mini-Circuits ZN2PD-9G + adjustable delay line
low pass filters	discrete 7^{th} order Butterworth, 10 MHz cut-off frequency

For the time being, a signal generator (HP 8665B) is used as a local oscillator, and a digital oscilloscope (Tektronix TDS7404) as ADC.

A photograph of part of the setup can be seen in figure 5. On the left, the LNA can be seen, mounted on a Rogers RO4003 substrate (white). It is connected to a power splitter. This is followed by the two mixers, again mounted on Rogers substrates. Another power splitter provides the LO signal to the two mixers. One can clearly see the different length of the connecting cables, resulting in a phase shift of 90^o (modulo 180) between the two channels. The mixers are both followed by a low-pass filter, which can just be seen on the top and top right of the photograph.

The receiver has been tested. This was done by applying test signals to the input of the front-end, using an Agilent E4438C vector signal generator. This generator can produce both Bluetooth and HiperLAN/2 signals.

Two input channels of a Tektronix TDS7404 oscilloscope were used as ADC's. The output data of the ADC's was imported into Matlab. In Matlab, the average noise power in

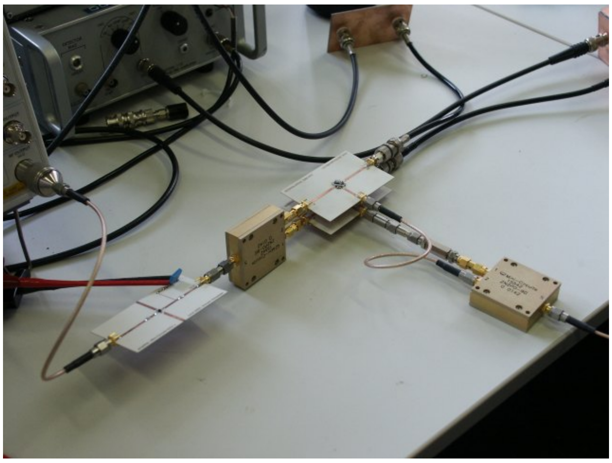
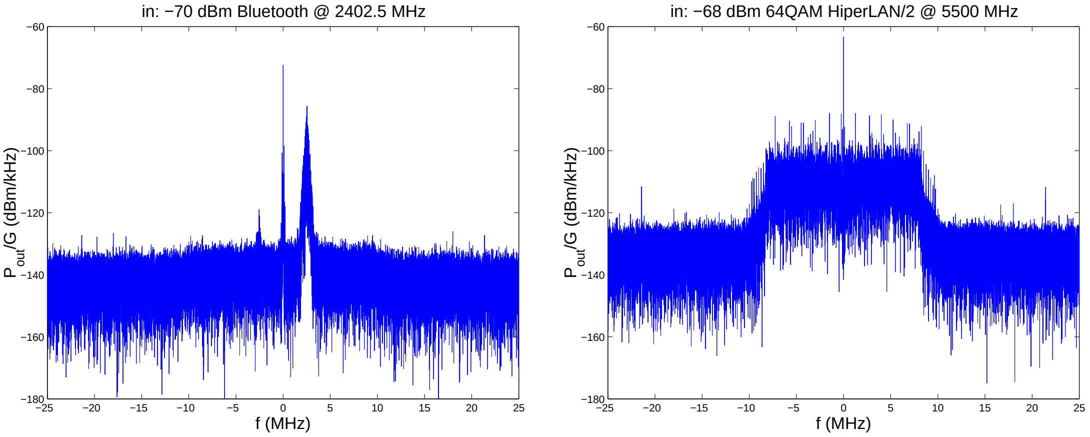


Figure 5: Photograph of part of the front-end



(a) Bluetooth input signal

(b) HiPerLAN/2 input signal

Figure 6: Measured output spectra of the receiver.

various parts of the spectrum was computed to determine the noise floor of the receiver. This is used to calculate the noise figure. The SSB noise figure at 2.4 GHz is 5.4 dB; at 5.5 GHz it is 14.5 dB. This includes the entire receiver, from LNA up to and including the ADC's.

Some other tests have been performed as well. Bluetooth and HiperLAN/2 signals were presented to the receiver, and the output signals can be seen in figure 6. These signals were also successfully demodulated on a general purpose computer. More information on these demodulation tests can be found in [8] and [9].

4 CONCLUSIONS AND FURTHER RESEARCH

A Software Defined Radio front-end test-bed has been designed. It works both as a Bluetooth and a HiperLAN/2 receiver. An important bottleneck for wideband receivers appears to be the linearity requirements, caused by strong out-of-band signals. This can be solved by using switchable filters.

As switchable filters impair flexibility of the receiver, an important subject of further research will be the front-end linearity of wideband receivers.

5 ACKNOWLEDGEMENT

We thank our colleagues from the Signals and Systems group for their work on the digital part of the front-end and for interesting discussions.

This research is supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW.

REFERENCES

[1] `http://nt5.el.utwente.nl/sdr/`

[2] Joseph Mitola III, "Software Radio Architecture", John Wiley & Sons, 2000.

[3] `http://www.sdrforum.org/sdr_primer.html#intro`

[4] Joe Mitola, "The Software Radio Architecture", IEEE Communications Magazine, vol. 33, no. 5, pp. 26–38, May 1995.

[5] ETSI, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer. 2001. ETSI TS 101 475 V1.2.2 (2001-02).

[6] Bluetooth SIG, Specification of the Bluetooth System - Core. Technical Specification Version 1.1, Bluetooth SIG, February 2001.

[7] Robert H. Walden, "Analog-to-digital converter survey and analysis", IEEE Journal on Selected Areas in Communications, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[8] L.F.W. van Hoesel et al., "Frequency Offset Correction in a Software Defined HiperLAN/2 Demodulator using Preamble Section A", MMSA 2002, *accepted for publication.*

[9] Lars van Mourik et al., "Performance Evaluation of a Combined HiperLAN/2-Bluetooth Digital Front-End", ProRISC 2002, *accepted for publication.*