A PLATFORM FOR THE DEVELOPMENT OF SOFTWARE DEFINED RADIO

Livia Ruiz, Gerard Baldwin and Ronan Farrell
Centre for Telecommunications Value-Chain Research
Institute of Microelectronic and Wireless Systems
NUI Maynooth
Ireland

ABSTRACT

This paper describes the development of an SDR (software defined radio) platform for use in investigating the requirements for implementing many popular standards such as GSM, WiFi and 3G through SDR. The first stage of this work is to generate specifications for an SDR implementation from various wireless standard requirements. The next stage is the development of suitable reconfigurable hardware. The final stage is the development of suitable software for the reconfiguration of the hardware and the implementation of the standards.

I INTRODUCTION

With the proliferation of modern wireless communication standards and the associated requirements for individual base stations with specific needs the demand for a flexible technology has increased. For instance it is common to find coverage for GSM, WiFi, 3G and WiMax in the same location provided by separate base stations for each standard. A more cost efficient solution could be provided by one unified base station. A technology capable of achieving this is SDR[1]. In an SDR solution the base station would be able to implement each standard by hardware and software reconfiguration.

This paper documents the development of both the software and hardware for a platform to experiment with an SDR implementation. The approach taken is, where possible, to use-off-the-shelf components with the software element implemented on a standard PC running the Linux operating system. Compared with using an FPGA or dedicated DSP processor this approach has the disadvantage of less computing power but it offers much greater flexibility.

This paper is divided into three main sections. Section II deals with the generation of the requirements for the SDR testbed from the specifications of the wireless standards. Section III gives in detail the development of the hardware design and its implementation. Section IV presents the software implementation.

II SDR REQUIREMENTS AND SPECIFICATIONS

A *RF Specifications*

The proposed platform is designed to operate in a frequency band from 1.6GHz to 2.5GHz and support the GSM1800, DCS1800, PCS 1900, UMTS-FDD, UMTS-TDD and 802.11b standards. The specifications for the SDR are then derived from these standards. In the above standards the specifications for the radio are characterized by the following parameters: sensitivity, noise figure and third order intermodulation for the receiver; and output power and phase noise for the transmitter[2]. The values specified in the standards for these parameters are shown in Table 1[3].

Table 1: RF Specifications for the Covered Standards.

Specifications \ Standards	**GSM**	**DCS 1800**	**PCS 1900**	**UMTS FDD**	**UMTS TDD**	**802.11**
Sensitivity(dBm / **SNR** (dB)	-102 / 9	-102 / 9	-102 / 9	-117 /18.9	-105 /6	-76.5 /10
IP3(dBm)	-18	-18	-18	-21.3	-20.9	-22.5
Noise Figure (dB)	9.8	11.8	9.8	9.6	9.2	11
Output Power (dBm)	39	36	36	33	24	11
Phase Noise(dBc/Hz)	-154	-154	-154	-148	-154	?

The sensitivity requirement for UMTS-FDD is the lowest at -117dBm, thus this is required sensitivity for this system. The IIP3 requirement for this platform is limited by the highest, which is for GSM and is : -18dBm. The NF_{total} must meet the required specifications of 9dB that arise from the GSM standard. The phase noise for the 802.11standard is not specified since this depends on the used modulation scheme. However, the phase noise requirement in 802.11 is more relaxed than the other standards. Hence, the hardest phase noise constraint is for GSM at -154dBc/Hz. The required maximum channel bandwidth is 22MHz, this arises from the 802.11.

B *Baseband Block Specifications*

The baseband block has to provide enough data bandwidth between the computer and the ADCs/DACs. It must also provide sufficient data resolution to correctly represent the signals. For this reason the speed and bit resolution of the ADCs/DACs have particular influence on the performance of the system. Each extra bit of resolution improves the SNR due to quantization error by 6dB. Extensive simulations were carried out to determine the optimum number of bits to use in the ADCs/DACs. From the communication standards selected, the most challenging constellation that this platform is required to support is OFDM (Orthogonal Frequency-Division Multiplexing). This is due to the large number of bits per symbol in this modulation scheme. As the bits/symbol increases the SNR requirements of the receiver increase[4]. This is reflected in the number of bits in the ADC. Thus, a constellation such as OFDM, will require

1-4244-1144-0/07/$25.00

more bits than simple BPSK (Binary Phase Shift Keying). For this reason OFDM is used as the worst case example in the simulations described here.

In the receiver the cyclic prefix of the OFDM signal is removed before reaching the uniform quantizer, which adds quantization noise to the signal. This is possible due to the assumption of perfect synchronization between the transmitter and the receiver. The BER (bit error ratio) is calculated through the SNR at the output of the system. Since the OFDM signal introduced into the system is a random process the SNR calculation requires the variance of both information and noise signals. To achieve the BER from the standards the required minimum number of bits in the ADC is sought. Figure 1 shows results from these simulations. These results are given as a function of the number of quantization bits. The number of quantization bits taken for the simulations was: 4, 5, 6, 7 and 8. It can be seen from this figure that at least 8 bits are required

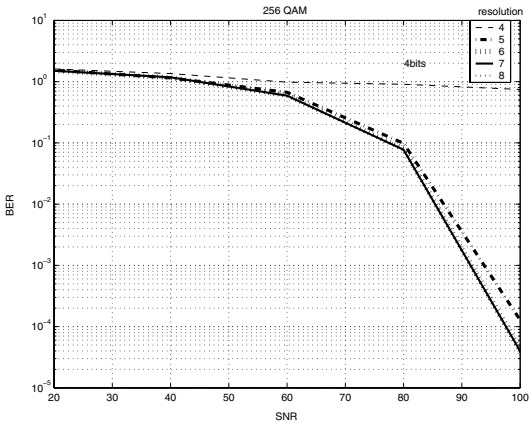


Figure 1: BER vs SNR for OFDM with 256QAM subcarrier modulation.

to achieve the necessary BER 10^{-3} with an efficient SNR in the system. Additional bits are added to allow for signal amplitude variation at the input of the ADC. Thus, 16-bit ADCs can be used which gives 8 bits of dynamic range to allow for amplitude variation in the incoming signal. This corresponds to 48dB of dynamic range available at the input to the ADC. This allows the relaxation of the requirements for the automatic gain control in the receiver.

As the maximum channel bandwidth is 22 MHz a sufficient sample rate is required to satisfy normal Nyquist sampling conditions. A 80Msps sample rate device is used to provide an extra guard bandwidth. However the sample rate of this part is controlled by software. Since 16-bit ADCs are used over a maximum channel bandwidth of 22MHz the total bandwidth results 352Mbps which is within the bandwidth range of the USB2.0.

A USB2.0 link is provided between the computer and the baseband system. This can operate up to 480Mbps. While this interface is not able to support the sustained data through-put of the system this problem is mitigated by the use of buffers in in the baseband block. Table 2 presents the resulting specifications for the design of this system.

Table 2: Specifications

Receiver	**Transmitter**	**Baseband**
IIP3:-0.87dBm	P_{out}:30dBm	ADC:16bit
Sensitivity:-104dBm	PN:-152dBc/Hz	Rate:80Msps
BW:22MHz	BW:22MHz	480Mbps

III HARDWARE DESIGN

As previously stated it is the intention to use as many off the shelf components as possible. The choice of a particular RF architecture to implement this platform is the key to the correct performance of the system. Three particular architectures were considered for the implementation of the SDR platform: superheterodyne, low IF, and direct conversion architecture. Even though the superheterodyne model is the most conventional architecture for wireless communication, the fact that RF and IF signals are processed by fixed analog components renders this architecture unsuitable for SDR purposes[2]. On the other hand, a Low IF architecture is a possible solution for SDR, however the requirement for an accurate image rejection filter transforms this into an unreliable architecture to use.

Although the direct conversion architecture has the disadvantage of DC offset, the image rejection filter is not needed. In addition the problems caused by the DC offset are easier to solve. Direct conversion requires low power dissipation, has easy tuning across large frequency bands, easy design, and lower cost because of the reduced number of components. This guarantees a higher level of integration than with other architectures[2]. Thus the direct conversion architecture has been chosen for this platform in both the transmiter and receiver. This technique directly transforms the RF signal to zero frequencies at down-conversion and from zero frequencies to RF at up-conversion. This approach has image suppression properties. Since the direct conversion architecture does not have problems with image signals, the pre-selection filter specifications at the receiver can be relaxed. Even though this architecture has many advantages compared with other ones, it has to deal with serious problems such as direct-current offset noise and I/Q mismatch[5]. Nevertheless the devices used in this system have been chosen in order to fulfill the requirements associated with these problems. Thus, the maximum I/Q amplitude imbalanced on the platform is 0.3dB with a maximum amplitude phase error of 1 degree and an IIP2 of 25dBm. The maximum baseband DC offset in the conversions stage is -42dBm.

The system is divided into three separate printed circuit boards for implementation. These three sections are : RF transmitter, RF receiver and baseband. Figure 2 illustrates a block diagram of the system.

A Baseband

The baseband board consists of the ADCs/DACs and USB interface. Figure 3, illustrates the printed circuit board layout of the baseband platform. This board has a size of 16cmx10cm.

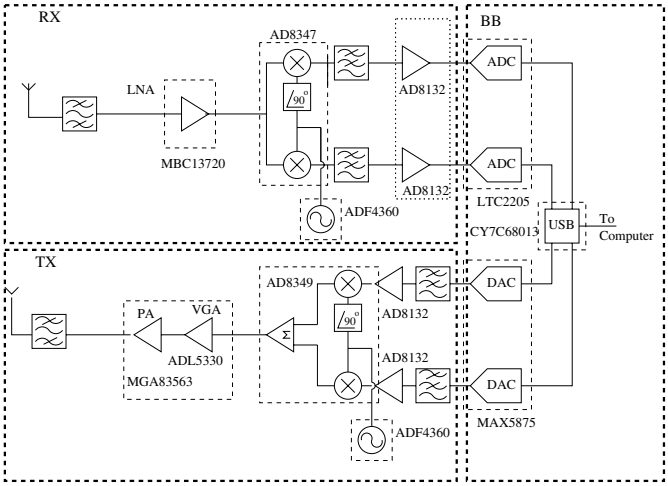


Figure 2: Block diagram of the system.

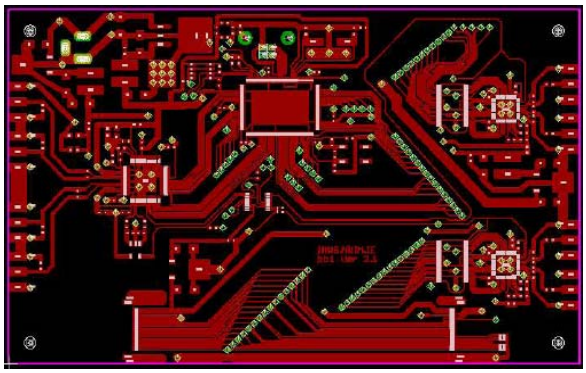


Figure 3: Baseband Board.

B *Transmitter*

On the transmission side the USB interface is followed by digital-to-analog conversion which is implemented by a MAXIM MAX5875 DAC. Similar to the receiver a pair of low pass filters are implemented using AD8132 opamps. These opamps feed the Analog Devices AD8349 up-converter followed by an Analog Devices ADL5330 VGA (voltage gain controlled amplifier). This VGA drives the Agilent MGA83563 power amplifier. The AD8349 up-converter gets its local oscillator signal from the Analog Devices ADF4360 synthesizer. An RF passband filter is implemented at the antenna using a coupled line microstrip filter.

The above elements of the transmitter are controlled from an 8051 microcontroller on the baseband board. The microcontroller's external 8 bit data and address bus is used to enable, disable and control the various components. This control architecture allows the various control pins to appear as memory locations to the microcontroller. This is achieved using a series of 8 bit comparators and latches.

Four different power supplies are used on this board: 2.5V, 3.3V, 5V and -5V. These are generated by on-board voltage regulators. The layout of the transmitter board is shown on figure 4. The transmitter board has a size of 16cmx10cm. The white outlined section contains the RF part of the transmitter board.

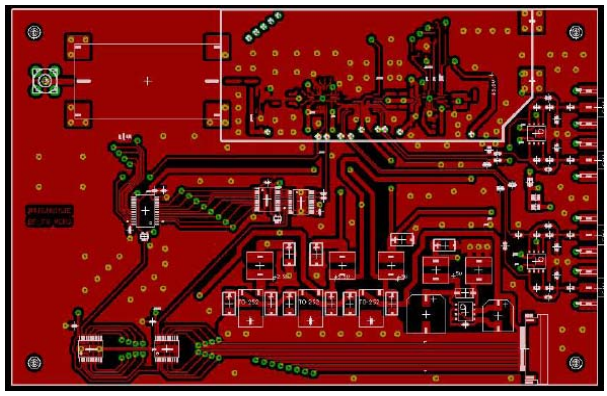


Figure 4: Tx Board.

C *Receiver*

The receiver is implemented using the Agilent MBC13720 LNA followed by the Analog Devices AD8347 downconverter which feeds a pair of low pass antialiasing filters implemented using the Analog Devices AD8132 opamps. An RF passband filter is implemented at the antenna using a coupled line microstrip filter[6], as in the transmitter section. The analog-to-digital conversion is performed by the Linear Technologies LTC2205 which are connected to the USB interface which is implemented using the Cypress Semiconductor CY7C68013.

Similarly to the transmitter, the microcontroller's external 8 bit data and address bus is used to enable, disable and control the various components. Four different power supplies are used on this board: 2.5V, 3.3V, 5V and -5V. The layout of the receiver board is shown on Figure 5. The receiver board has a size of 16cmx10cm. The white outlined section contains the RF part of the receiver board.

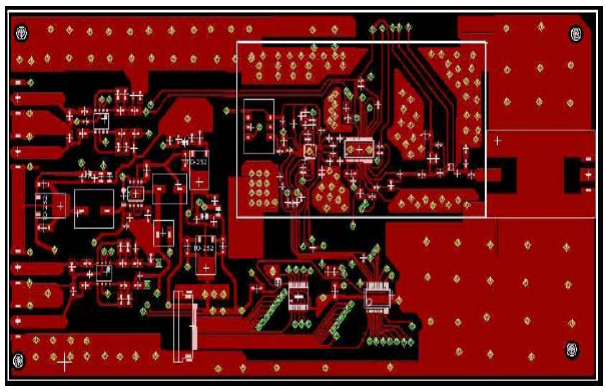


Figure 5: Rx Board.

IV Software Design

The are three different software stacks implemented in this platform.

Microcontroller stack:

This code, written in C, implements the embedded control functions such as the frequency synthesizer centre frequency selection, VGA power control and device enabling and disabling. It also manages the baseband board side of the USB interface. The compiler used is the SDCC (small device C compiler). The compiled binary file is transferred to the microcontroller over the USB interface at USB connection time. This allows the updating of the embedded firmware to be updated each time the USB connection is made.

PC driver:

This code implements a be-spoke USB driver for the device. This driver provides separate USB end points for configuration control, data input and data output. The configuration information is transferred in 64 byte packets to the microcontroller memory locations stated earlier, enabling direct control of hardware components from the PC. The data is transferred in 512 byte packets, to and from FIFOs on the microcontroller side, to and from buffers on the PC side. The control and data interfaces are accessed on the PC side through a pseudo-file.

System software:

This consists of an interface between Matlab and the pseudo-file written in C. This interface permits the implementation of communication schemes directly in Matlab.

Currently the microcontroller stack and PC driver are written. The Matlab interface is completed, however as yet only simple communication schemes have been implemented in Matlab.

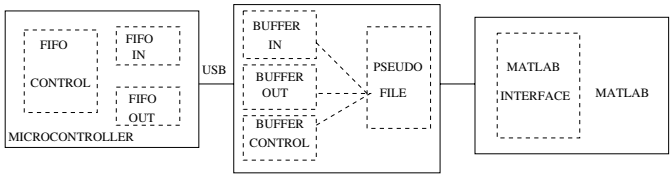


Figure 6: Software Stacks.

V RESULTS

Results are presented for simple BPSK and QPSK communication schemes using the SDR platform. In order to obtain these results the demodulator of this SDR platform has been sampled periodically to recover the information signal. For a synchronously sampled symbol, timing has to be derived from the received signal since the propagation delay is unknown at the receiver[4]. Thus, timing recovery has been performed in the PC. For early simulations, the Simulink Matlab environment has been used. However a timing recovery application has been written as part of the platform software stacks. Results for both the system simulated using Simulink and the software code written for the platform are illustrated. Figure 7 shows the receiver implemented in Simulink.

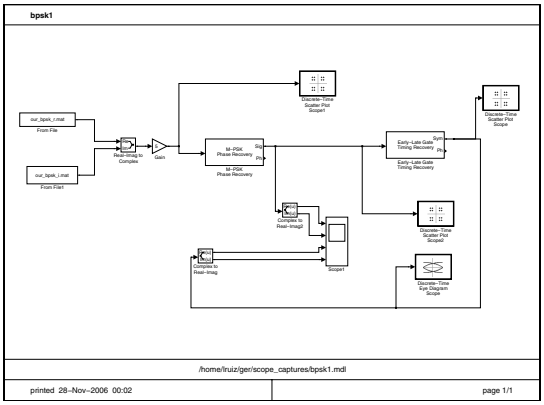


Figure 7: Matlab implementation.

The transmitted signal consists of raised cosine pulse with 50% rolloff at 20kbps modulated on to a carrier at 2.21GHz using BPSK and QPSK respectively.

The receiver implements non data added carrier and timing recovery. Figure 8 shows the constellation for a received BPSK signal after timing and carrier recovery. Figure 9 shows the constellation for a received QPSK signal again after timing and carrier recovery. Figure 10 gives the eye diagram for the received BPSK signal. Figure 11 shows an oscilloscope capture of the received signal in the BPSK case before digitization. The carrier recovery software stack is performed us-

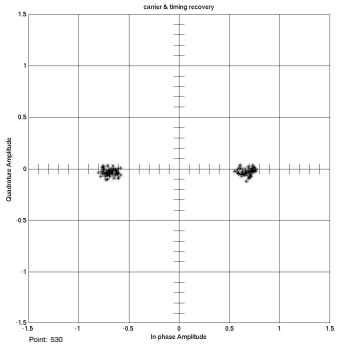


Figure 8: Constellation for a BPSK received signal.

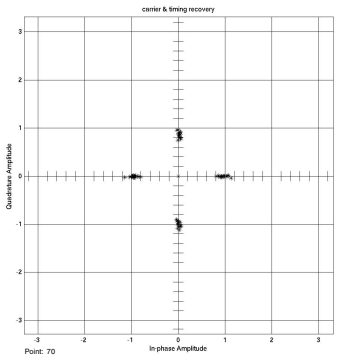


Figure 9: Constellation for a QPSK received signal.

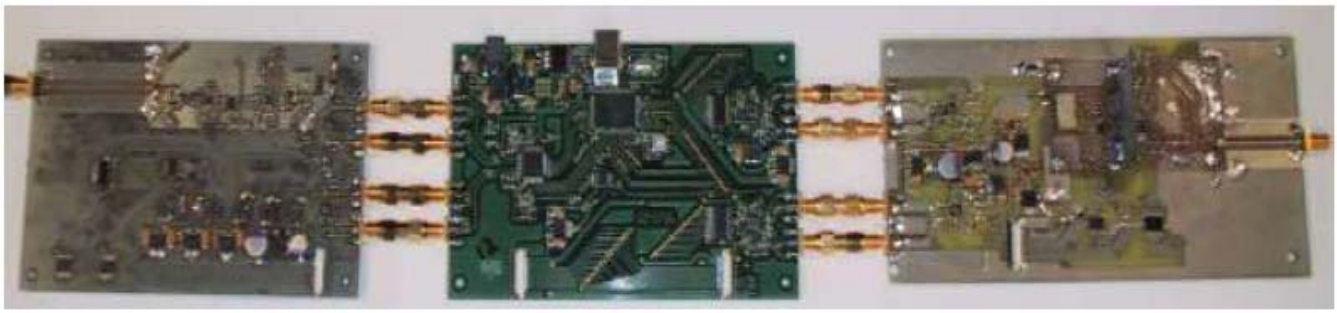


Figure 13: SDR Platform.

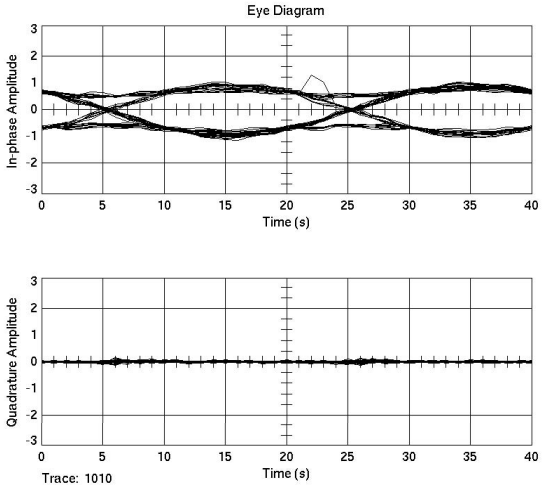


Figure 10: Eye diagram for a BPSK received signal.

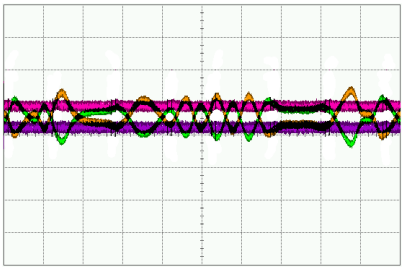
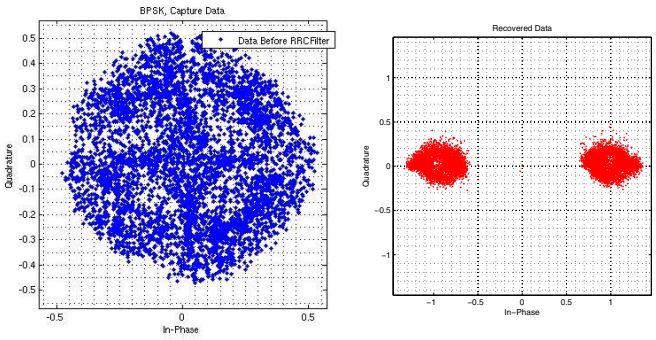


Figure 11: Received Signal in the Oscilloscope.

ing the decision-feedback phase-locked loop(DFPLL). A root raised cosine filter and a carrier recovery system are modeled in Matlab in order to estimate the phase error of each sample and recover the original carrier phase. Figure 12 shows the constellation of both the unprocessed received data and the processed BPSK signal after timming recovery. A photograph of the hardware elements of the platform is shown in Figure 13

VI Conclusion

This paper has presented a testbed platform for experimentation with SDR technology. It has used off the shelf components in both hardware and software. It has been focused on the direct conversion architecture. Hardware and software sections of the SDR have been described and preliminary results provided.



(a) Constellation of BPSK before Timing Recovery.

(b) Constellation of BPSK after Timing Recovery.

Figure 12: Carrier recovery

Acknowledgment

The authors would like to thank the Center for Telecommunications Value-Chain-Driven Research(CTVR) and the Science Foundation of Ireland (SFI) for supporting this research.

References

[1] Jorge M.Pereira "Reconfigurable Radio: The evolving Perspectives of different players". Personal, 12th IEEE International Symposium on Indoor and Mobile Radio Communications, 2001

[2] Tuttlebee Walter "Software defined radio:enabling technologies", 2002.

[3] B.Razavi. "RF Microelectronics". Prentice Hall,1998.

[4] J.G.Proakis "Digital Communication,4th ed.",2001.

[5] P.Zhang, T.Nguyen. "A 5-GHz direct-conversion CMOS transceiver". IEEE Journal of Solid-State Circuits.

[6] George L. Matthaei, Leo Young, E.M.I.Jones "Microwave filters, impedance-matching networks, and coupling structures". Artech House, ISBN 0890060991, 1980.