A New Architecture for Wireless Smart Sensor Based on Software Defined Radio

P. Ferrari, A. Flammini, E. Sisinni
Dept. of Electronics for Automation
Faculty of Engineering, University of Brescia
Brescia, Italy
paolo.ferrari@ing.unibs.it

Abstract—Today wireless sensors technology is based on monolithic transceivers that optimize cost but have rigid hardware architecture. In this work, a new architecture for wireless sensors is presented. It is based on the Software Defined Radio concept and shows an impressive adaptability to the external conditions. The proposed architecture, called Wireless Ultra Smart Sensor, enables the use of a software programmable transceiver which can simultaneously manage multiple radio communication channels or standards. In addition, the Wireless Ultra Smart Sensor architecture can mix the flexible communication part of the smart device with the sensor conditioning part. This combination represents a new generation of general purpose sensors for microwave measurements in the GHz range (e.g. displacement and position sensing) with enhanced versatility, since the same radio frequency front-end is used both to communicate and to make measurements. The experimental results demonstrate the feasibility of the proposed architecture. The prototype shows the abilities of: managing multiple protocols using a single antenna; interfacing standard sensors; making delay and phase measurements with a standard deviation of 3 ns; and assigning time of arrival to incoming radio signals with an error on the order of 20 ns.

Wireless sensor; Software defined radio; Synchronization; Time of flight; Performance evaluation.
I. INTRODUCTION

The Wireless Sensor Networks (WSNs) are becoming very popular in both industrial and home automation applications. Unfortunately, each technology provider generally uses proprietary, low-cost wireless solutions for its communication and networking, making the current situation very fragmented. For instance, some industrial applications [1] use IEEE802.15.1 technology, whereas the research in automotive is investigating to adapt IEEE802.11-like technologies to wireless sensor networks [2, 3]. Other important problems are due to the fact that the communication modulation and protocol are usually hardware-coded inside the transceiver, i.e. the physical device appointed for baseband signal processing and modulation. A typical example of this matter is given by the case of ZigBee (for home/building automation) [4], WirelessHART [5], and ISA SP-100 (for process and factory automation) [6]. Although these three standards are based on IEEE802.15.4, some differences (e.g. frequency agility or power adaptation) have been introduced to fulfill specific application requirements; as a result, the use of a common transceiver is quite difficult.

Moreover, many coexistence problems [7] are arising among different communication standards, which are used by WSNs operating in the same area. Even in telecommunications field, more standards coexist in products like cellular phones (GSM, UMTS, WiFi, and BT) and the concept of multi-standard radio is emerging. A lot of research work has been conducted about Cognitive radios [8], which are able to dynamically adapt the communication protocols stack, down to the modulation scheme, according to the system needs. Cognitive radio concept is based on Software Defined Radio (SDR) architecture [9]. A SDR uses a simple front-end structure in which the analog signals are converted into digital signals as soon as possible (very close to the antenna), while standard functionalities of a communication interface (processing, demodulation, protocols stack) are performed by software, giving extreme flexibility to the system.
Obviously, the implementation of a device according to the SDR architecture has a higher cost than the traditional transceiver-based solution. However, the decreasing cost of electronic components (especially if adopted by a large-scale market like consumer electronics or automotive) and the increasing requirements of wireless sensor networks in terms of reliability and versatility, make this architecture appealing for the near future [10, 19].

This paper proposes a new architecture for smart sensors in order to improve communication flexibility and to provide some sensing functions, taking advantage from the same circuits used for communication.

The first new idea introduced in this paper is the communication flexibility of wireless smart sensors. The proposed smart sensor communicates using a particularly versatile hardware that changes its behavior by means of software reprogramming. The new smart sensor exploits SDR architecture for implementing a dynamically adaptable transceiver, capable to exchange messages with different, concurrent, partners. This extreme flexibility allows a sensor to belong to different networks, making its information more accessible and reliable.

Besides, the interest on microwave-based measuring systems, which is always high in some specialized sectors like military or space applications, is now increasing in the general purpose sensor market. For instance, devices for industrial or civil use, like small RADARs, displacement sensors, and localization apparatuses, employ electromagnetic waves in the GHz range to achieve their measurements. Especially in the low-end systems, the introduction of new concepts of flexibility and adaptability could boost new applications; the possibility to change the measuring method according to environment variations could lead to a new generation of general purpose sensors with enhanced performance.

Since the introduction of the concept of scalable multifunction RF systems in [11], several researches were focused on the integration between high-end RADAR and communication functions in a single RF system with the aim of reducing weight and cost for military applications [12, 13]. Such architectures (also
known as multifunction RADAR) are based on very powerful measurements platforms (i.e. military grade RADAR) to which the communication features are added.

On the contrary, in general purpose (low-end) wireless sensors with microwave measurement capabilities, two separate devices (with different antennas) are normally used: a transceiver for communications; and a specific circuit (or a device) realizing the selected measurement technique (e.g. time of flight, resonance for a cavity, etc.) Thus, the second idea behind this paper is a new sensor architecture, which shares the same radio hardware interface between the SDR-based communication system and the sensing system. For example, a sensor could use RADAR techniques to measure distances and then it could transmit the result using the same radio front-end; both operations could be done maximizing coexistence with neighboring sensors.

The architecture proposed in this paper shares some concepts with the multifunction RADAR but it is strongly oriented to general purpose wireless smart sensors build around low-end SDR platforms. Up to now, the low-end SDR platforms are usually employed for telecommunication applications [14] and rarely for RADAR systems [15]. When measuring functions are considered, the focus is either on software aspects or on specific processing and measuring algorithms, not on the shared architecture between measurement and communication. Last, the architecture proposed in this paper has the capability to refer its measures to an accurate time base, which is a fundamental requirement for sensors that use radio waves to make measurements (e.g. phase measurements, frequency measurements, time-of-flight measurements, triangulation).

In this paper, the feasibility of the proposed architecture, called Wireless Ultra Smart Sensor (WUSS) [16], is evaluated using a standard SDR platform. In particular, the architecture of a new type of wireless smart sensor, capable to communicate with different partners at the same time, is proposed and tested. The time characteristics of the prototype are investigated; local oscillator stability and uncertainty, introduced when making delay or phase measurements, are estimated. Last, the capability of the proposed WUSS to
acquire signals with precise time reference (e.g. timestamping) without using special hardware or proprietary protocols is evaluated. In the following, a brief overview of SDR concepts is given; then, the feasibility is demonstrated by means of working hardware prototypes and experimental tests.

II. ARCHITECTURE OF A SDR

Fig. 1 shows the high-level diagram of a generic SDR: there is a radio frequency front-end followed by a stage of broadband conversion; finally the digital signal enters in (or exits from) a generic data-processing system. Turning from abstraction to practical implementation, generally some characterizing "key blocks" can be identified in a SDR, as shown in details at the bottom of Fig. 1.

A low-noise amplifier is placed immediately after the antenna, followed by a section of "IF managing": this block deals with down-conversion and up-conversion, translating the frequency of interest into a generic intermediate frequency (IF) and vice versa. The IF block is usually software-configurable, for instance, setting the center frequency. Moreover, some filters could be inserted in this block to reduce the bandwidth of interest in a range that agrees with the next block of sampling/generation (in order to satisfy Nyquist's Theorem). Obviously, the IF block should be as simple as possible, since the most of the signal processing is

Figure 1. Block diagram of a generic Software Defined Radio.
numeric. Conversion section is the most important one of the chain, being the block that determines the bandwidth of the entire system. The A/D converter fixes the range of frequencies within which the SDR will work; it should be positioned as close as possible to the antenna to optimize performance. After the conversion block, the signal samples can be processed by a generic processing unit (a DSP, a microcontroller, an FPGA or a general purpose PC), giving extreme flexibility to the system. In the SDRs available on the market, a hybrid structure is implemented in which a programmable logic (e.g. FPGA) manages high speed repetitive tasks, while a processor (generally a PC) deals with higher levels of the communication stack.

III. THE PROPOSED APPROACH

A traditional wireless sensor node is shown in Fig. 2a. The transceiver is a block that transforms digital data into modulated RF signals and vice versa; it realizes in hardware the most of functionalities of the physical and MAC layers, including modulation schemes. As a result, a traditional wireless sensor has rigid hardware architecture and it is able to communicate only with a single protocol even if several other protocols are sharing the same RF band (e.g. ISM - Instrumentation, Scientific, and Medical band).

In this paper, in order to take advantage from the SDR solution, a new architecture for wireless sensors is introduced in two steps. First, the transceiver of Fig. 2a has been substituted by a SDR-based architecture that implements a flexible transceiver, as illustrated in Fig. 2b. The “software” transceiver is able to modify its behavior through a simple software reprogramming, thus it is not tied to a particular communication protocol. As a result, the new flexible wireless smart sensor has advanced transmission features inherited from the SDR [17]: modulation adaptation, spectrum analysis capabilities, radio signal preprocessing/optimization, and so on. Hence, the proposed architecture shown in Fig. 2b is able to adapt its communication protocol and modulation to the environment, and to simultaneously use multiple channels or multiple standards,
introducing redundancy and maximizing the probability of communication success. The architecture of Fig. 2b could be used to connect any type of transducer (e.g. temperature, flow, pressure, etc.).

The next step leads to the Wireless Ultra Smart Sensor (WUSS), i.e. the new proposed architecture for wireless sensors. The further improvement proposed in this paper is to use the SDR-based architecture for offering both transmission features and measurements capabilities, as shown in Fig. 2c. For instance, this architecture can be used for microwave sensors, which operate in the same RF band of the communication protocols, to make phase measurements, time-of-flight measurements and so on. Once again, the combined functionalities are provided using the “software” transceiver that can be reprogrammed to become the measurement probe.

In order to practically realize such a kind of reconfigurable systems, a suitable platform must be found. Two constraints must be considered: during the preliminary characterization, the system should have easy software programmability and high computational power; during the optimization phase the system should offer high scalability and the possibility to use programmable logic (e.g. FPGA). In the first phase, the objective is the creation of suitable software algorithms to enhance flexibility, whereas in the second phase the goal is the cost/size reduction still maintaining adequate performance.

The starting point for demonstrating the feasibility of the proposed approach can be an already-available SDR platform. Indeed, the authors aim is not to realize an engineering version of the WUSS, but to test and verify the architectures described in Fig. 2b and 2c. Such goal can be easily reached adopting a general purpose commercial-available platform. In the next sections, a reference SDR platform is chosen and its characteristics are evaluated. It should be said that many SDR platforms are already well described from the communication point of view but, in this paper, special attention is paid to the evaluation of the time related characteristics (accurate timestamping of events and synchronization capabilities), as previously highlighted in the introduction. As last consideration, the resulting prototypes are based on the combination of an FPGA
and a powerful processor, implying higher power consumption than traditional smart sensors. However, the technology is continuously evolving and low power solutions are rapidly arriving in the market (for instance IGLOO FPGA from ACTEL and ATOM processor from Intel).

![Block diagrams](image)

Figure 2. a) Block diagram of a traditional wireless smart sensor; b) the architecture for a new SDR-based wireless sensor; c) the architecture of the proposed Wireless Ultra Smart Sensor (WUSS).

IV. THE REALIZED PROTOTYPES

Nowadays, there are several implementations of the concept of SDR, each one customized for a single application and its related needs. Surely, the most powerful SDR platforms in terms of performance are the commercial ones; in this group can be included products of manufacturers such as Sundance, Pentek, and Texas Instruments. In addition to these solutions, even non-commercial projects can be found: these implementations (that follow an open source philosophy) are Linrad, HPSDR, and GNU Radio. In particular, GNU Radio has been chosen for the prototype systems realized in this paper.

GNU Radio is an open source project to create SDR composed by a minimal hardware (development-oriented) and a modular and expandable library that includes function blocks for signal processing. Software is block-organized and is composed by a hybrid C++/Python system: signal processing primitives are written
in C++ while interfaces between blocks are written in Python, a dynamic object-oriented scripting language. Essentially, Python connects processing blocks allowing data to flow at maximum speed without being interpreted. Integration of C++ into a scripting language is provided by SWIG, an interface compiler. Many signal processing blocks are already developed and are available to GNU Radio community to facilitate the developers’ work.

As previously said, GNU Radio needs some radio frequency front-end hardware: several different hardware platforms are supported, but the most commonly used is the Universal Software Radio Peripheral (USRP). USRP is a USB2.0 device, developed by Ettus Research LLC specifically for the GNU Radio project. It is a low-cost and high-speed equipment interfacing a general purpose PC to the radio frequency world. USRP consists of a motherboard holding four 12-bit ADCs (up to 64 MS/s sampling rate), four 14-bit DACs (up to 128 MS/s) and an Altera FPGA for simple high speed operations as up-conversion, down-conversion, interpolation and decimation. ADCs and DACs allow to receive baseband signals up to 32 MHz and to generate baseband signals up to 50 MHz. The frequency band considered in this paper is the 2.4 GHz ISM band. In particular, on the USRP motherboard, the daughterboard RFX2400 has been used for tuning such frequencies. In details, the daughterboard hosts two radio frequency analog mixers that are programmable with steps of 4 kHz ranging from 2400 to 2700 MHz.

An SDR according to the GNU Radio project and based on USRP is a low-cost system with some drawbacks. Considering the industrial sensor applications, the first limitation is that the GNU Radio project is currently oriented to transmission of continuous stream of data. It does not provide good support for packet protocol processing. Today, there is an ongoing structure renovation with the development of a new primitive, the so called “m-block”, which can facilitate the implementation of packet based processing and the annotation of samples with metadata (e.g. the timestamp, which is the time instant related to a packet transmission/reception). However no stable version is available, yet. In addition, there is the bottleneck of the USB2.0 based interface of the USRP. Even if the theoretic limit is 480 Mbit/s, in practice the USB2.0 chip of
the USRP can sustain about 32 Mbyte/s of continuous data throughput; this is a quite strong limitation. Generally, the RF signal is translated to a lower frequency by means of a quadrature modulator (i.e. is multiplied for two sinusoidal signals with a 90° phase offset) obtaining two outputs called In-phase output (I) and in-Quadrature output (Q). Sampling the I and Q components generates a two elements vector that can be considered as a complex number, where the amplitude of I signal is the real part and the amplitude of Q signal is the imaginary part. It is important to highlight that all samples sent over the USB interface are in 16-bit signed integers: in other words, the use of this IQ format (16-bit I and 16-bit Q samples) limits the transfer rate across USB to a maximum of 8MS/s of complex samples.

Although these drawbacks, our choice is justified by the aim to experimentally evaluate the proposed architecture. The structure of the realized prototype of the proposed WUSS (see, for instance, Fig. 3) is very far from a real sensor concerning size, cost, and power consumption, but we hope that the technology trend will allow to replace the PC and the FPGA with a single low-power, low-cost device according to architectures shown in Fig. 2b and Fig. 2c.

V. EXPERIMENTAL SETUP AND RESULTS

As discussed in the introduction, the characterization is aimed to assess the feasibility of the proposed architectures; clearly the results depend also on the chosen SDR platform. In particular, two main objectives and several sub-objectives can be individuated:

1. evaluation of the architecture of Fig. 2b. In particular, (section A) the flexibility of the prototypes will be assessed evaluating the abilities:

   - of accessing sensor information using modulation schemes and protocols chosen among the most diffused ones in the sensor application field

   - of simultaneously using multiple modulation schemes, for instance managing two different clients that simultaneously interrogate the WUSS with two different protocols. For the sake of
completeness, a temperature sensor has been included in the setup, in agreement with the architecture shown in Fig. 2b.

2. performance evaluation of the architecture of Fig. 2c. In details:

- the hardware front-end (i.e. the USRP) will be evaluated in terms of time characteristics; the local oscillator behavior will be evaluated (section B)

- the stability and repeatability of phase measurements, made using the hardware front-end (PC, FPGA-based processing, DAC and ADC), will be evaluated (section C). This feature is essential to make measurements like time-of-flight or to realize RADAR-like systems.

- the synchronization accuracy of the WUSS prototype will be estimated considering two experimental setups and three different situations (sections D, E, F). The latency and jitter (i.e. maximum variation of the time latency) in assigning a timestamp to modulated data coming from external sources are used as performance indicators. These experiments are aimed to demonstrate that the prototype of WUSS can refer the sampled data to an accurate time base.

A. Evaluation of the architecture flexibility

This section demonstrates that the architecture described in Fig 2.b is feasible by means of two experiments. The first part is the demonstration that the prototypes can exchange data using data packet protocols. The Packet Error Rate (PER) of the sensor prototype has also been evaluated. The system developed to implement the first test on transmission and reception of data packets involves two communicating GNU radios and a spectrum analyzer (RSA3408A – Tektronix), as shown in Fig. 3. The communication functionalities have been evaluated using both freely available modulation schemes, coming with the GNU Radio package, and the IEEE802.15.4/Zigbee module from UCLA [18]. The transmission and reception of data packet have been successfully tested and the PER of the link between the two GNU Radios has been computed. The results show that the PER increases with the distance in the range from 1 m to 20 m
(the maximum is 10%). The phenomenon is principally due to the high noise introduced by active Wi-Fi links that share the same spectrum in the same area of the test, as shown in Fig. 3.

![Figure 3. Experimental setup for the evaluation of data packet transmission and reception.](image)

The second experiment points out the flexibility of the proposed system. The experimental setup is shown in Fig. 4. There are two wireless nodes using two different modulation schemes (proprietary GMSK-based, center frequency 2427MHz; and standard IEEE802.15.4, center frequency 2420MHz) that, exactly in the same time instant, ask the proposed smart sensor for the temperature measurement. In the Fig. 5 it is shown the spectrogram computed by the RSA3408 in the region from 2404 MHz to 2440 MHz, highlighting the actual time overlapping of the two packets. The test goal is to correctly receive both messages (even if overlapped in time) answering to both the nodes using their respective modulations.

The architecture of Fig. 2b has been realized adding a temperature sensor to the USRP hardware; in details, the temperature sensor is a TMP102 attached to the I^2C bus available on the USRP board as shown in Fig.4. Then, the SDR part of the proposed smart sensor has been realized creating software with a double transmission path and a double reception path. As a result, the sensor node listens in the RF bandwidth (2.415 GHz to 2.430 GHz in this experiment), concurrently seeking for requests coming from the two nodes, i.e. the RF signal samples are demodulated by the software in two different ways. If a request is detected by a
reception path, the front-end commutes into transmission state and the corresponding modulation is used to send the response. The test results show that the WUSS prototype can simultaneously deal with the two different communication standards taken as an example, since the percentage of requests without response is on the same range of the PER of the previous test. The tested capability cannot be generalized to every communication standard or environment condition, due to GNU and USRP limits but, as it was said, the realized prototype has been used just as a proof-of-concept.

It should be noted that if traditional wireless sensor architecture is used to fulfill the requirements, a low-flexibility architecture with two transceivers should be realized.

Figure 4. Experimental setup for the flexibility test of the proposed system.
B. Evaluation of timing characteristics

The aim of the proposed WUSS is also to make measurements with the same front-end that is used to transmit and receive data. Hence, the evaluation of timing characteristics related to the front-end is fundamental to set the upper limit to the synchronization accuracy. Time delay and jitter of data processing after the USRP vary according to the computational power of the PC and can be reduced increasing the PC computational power, while uncertainty introduced by the front-end cannot be reduced.

The first test consists of the evaluation of the local oscillator used in the prototypes. The RF front-end includes a single local oscillator that feeds both the main board USRP (where lies the FPGA) and the daughterboard. The nominal value of the oscillator is 64 MHz with a declared accuracy of 10 ppm. A precision counter (Agilent 53123A with high stability option) has been used for measuring. The frequency measures in the short period are made at room temperature by setting the gate to the following values: 1 ms, 10 ms, 100 ms, 1 s. For each value, 100 measures have been taken and results for two front-ends are shown.

Figure 5. Spectrogram of the two simultaneous IEEE802.15.4 and GMSK transmissions.
in Table I. In conclusion, the short term stability of the oscillator is fairly adequate for stable carrier generation and also for short interval time measurements.

TABLE I. STABILITY OF THE LOCAL OSCILLATOR OF TWO FRONT-ENDS.

<table>
<thead>
<tr>
<th>Gate time</th>
<th>USRP 1</th>
<th>USRP 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average (Hz)</td>
<td>Std. Dev (Hz)</td>
</tr>
<tr>
<td>1 ms</td>
<td>63 999 436.9</td>
<td>6.8</td>
</tr>
<tr>
<td>10 ms</td>
<td>63 999 437.5</td>
<td>1.30</td>
</tr>
<tr>
<td>100 ms</td>
<td>63 999 437.2</td>
<td>0.17</td>
</tr>
<tr>
<td>1000 ms</td>
<td>63 999 437.2</td>
<td>0.35</td>
</tr>
</tbody>
</table>

C. Stability and repeatability of phase measurements

The ability to make accurate delay or phase measurement between transmitted and received signals is crucial for any application that needs to calculate, for instance, the time-of-flight in reflection-based measurement techniques. In order to evaluate the contribution to stability and repeatability of the USRP motherboard only, a hardware loopback has been created between DAC output and the ADC input. Since the local oscillator is the same for the transmission and the reception path, the use of the same board to transmit and receive signals, avoids any error introduced by the local oscillator variations.

The test signal is the continuous repetition of sinusoidal bursts; each burst is composed of 10 sine periods followed by a quiet time. The sine frequency is 800 kHz, the generation and the sampling frequency is 4MS/s (maximum limit with the USRP hardware), and the burst length (sine periods plus quite time) is 250 samples (or 62.5 µs).
The goal of the experiment is the characterization of the delay between generation and reception of each burst. The transmitter-to-receiver delay is calculated executing the cross-correlation between the received and the transmitted sampled signal. In order to improve the resolution beyond the 250 ns for the detection of the cross-correlation peak, a barycentric interpolation over 5 points has been applied.

The first result of the experiment is that the transmitter-to-receiver delay greatly changes among different runs (i.e. switching off and switching on the test signal generation): the mean value of the transmitter-to-receiver delay is 1136.04 samples (284.01 μs) and the standard deviation is 634.32 samples (158.58 μs). The reason is the greatly varying delay of the Operating System and of the USB hardware before establishing a connection with the USRP hardware.

On the other hand, in the same run (i.e. without switching off the transmitter) the transmitter-to-receiver delay exhibits a very stable behavior. The first received burst can be used as the reference for calibration of the transmitter-to-receiver delay for the current run. In Fig. 6 (realized measuring the delay of 4000 bursts), the distribution of the error with respect to the reference is shown: the standard deviation is 0.0119 samples (2.98 ns).
Figure 6. Distribution of the transmitter-to-receiver delay error after the calibration (4000 events).

In conclusion, the good stability, due to the single oscillator of DAC and DAC, allows a future implementation (out of the scope of this paper) of a measurement session composed by two phases: first a loopback path is enabled and the synchronization between transmitted and received sampled signals is achieved (e.g. by cross-correlation); then the loopback is disabled and the delay due to reflection can be evaluated.

D. Sinusoidal carrier modulated by a RF generator

Another important characteristic of the proposed architecture is the time synchronization accuracy using wireless methods, i.e. the ability of sharing the same time reference even if the systems do not use the same oscillator. Typical applications involve several nodes realized by means of different hardware platforms, which use standard protocols to exchange information. The WUSS synchronization should work also in this scenario, thus the objective of the following experiments is to evaluate the synchronization accuracy obtained exploiting easy solutions. The main constrain is the use of simple signals like sinusoids or modulated communication frames.
For this reason, in this paper, the time synchronization characteristics of the front-end have been initially evaluated using ON-OFF modulated sinusoidal signals (this section and section E). As the last step (section F), the measurements have been done using a packet transmitted by a commercial transceiver.

The RF signal appears to the receivers after some time that the modulation pulse has been applied to the RF source. Such a delay is composed of four parts: delay inside the radio frequency source, time-of-flight, delay in the USRP analog mixer, and processing delay inside the FPGA. In the experiments, the delay can be estimated “a priori” so the performance indicator is the error between the average measure delay and the expected delay. It is not eliminable and will affect all the measurements derived from the delay estimation (e.g. not only synchronization, but also time-of-flight between nodes, localization using emitting anchor nodes, etc.). A simple way to estimate the delay is measuring the time difference between the rising edge of the transmit trigger $S_m$ and the instant the received signal by WUSS becomes greater than a fixed threshold. In all the following experiments the main sources of uncertainty are: the noise on the analog signal that can affect the threshold crossing; the time quantization in capturing both the crossing instant and the signal $S_m$; and the latency of the RF generator/transmitter. The related effects can be reduced averaging over several measures.

The first experimental setup is shown in Fig. 7. In addition to the WUSS there is an RF generator (Agilent E4422B) used as the radio frequency source. It is programmed in ON-OFF keying and it is controlled by the modulation generator (Agilent 33220A) with the signal $S_m$. In all the experiments (including sections D, E and F), the FPGA of the USRP is loaded with modified firmware called "gr-GPIO", which activates the sampling of a digital input (routed to $S_m$) in parallel with the digitalized analog signal coming from the mixer stage.
The modulation frequency is 250 Hz with duty cycle of 25%. This means that ON pulses last 1 ms and are repeated every 4 ms. Signals acquired by the USRP on I and Q channels are shown in Fig. 8. For sake of completeness, the digital signal $S_m$ is also visible. In particular, the transmitted signal has a frequency of $f_{\text{sin}}$ and the USRP is tuned to the frequency $f_{\text{ric}}$, slightly different from $f_{\text{sin}}$. The observed sinusoidal signal at the receiver has a frequency $f_{\text{demod}} = |f_{\text{sin}} - f_{\text{ric}}|$.

The results are shown in Table II varying the sample rate and taking 1000 measures. The average value of the measured delay is reported, together with the standard deviation and the jitter (i.e. the difference between the maximum measured value and the minimum measured value). The generator and the analog part of the front–end introduce about 918 ns of constant delay. The internal architecture of the FPGA includes some digital filters and FIFOs; hence signals are delayed of a number $N_{\text{delay}}$ of samples that depends only from the sampling rate. The exact expression for this delay is $N_{\text{delay}} = 10.5 + ((D-2)/D)$, where $D = 64 \text{ MHz/Sampling\_rate}$ is the decimation factor. The value of the time-of-flight is negligible in the current experimental setup, since the distance between the transmitter and the receiver is fixed to 1 m. The value $\text{Exp} = 918 \text{ ns} + N_{\text{delay}}/\text{Sampling\_rate}$, which is visible in Table II, indicates the expected delay using this information. The absolute error of the average value with respect to the expected value is always less than the value of the standard deviation, confirming the measure validity.

**TABLE II.** MEASURED DELAY (µS) IN THE CASE OF A SINUSOIDAL CARRIER MODULATED BY A RF GENERATOR.
<table>
<thead>
<tr>
<th>Sample rate</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Jitter</th>
<th>Exp.</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 MS/s</td>
<td>2.346</td>
<td>0.060</td>
<td>0.375</td>
<td>2.324</td>
<td>0.022</td>
</tr>
<tr>
<td>4 MS/s</td>
<td>3.765</td>
<td>0.063</td>
<td>0.500</td>
<td>3.761</td>
<td>0.003</td>
</tr>
<tr>
<td>2 MS/s</td>
<td>6.585</td>
<td>0.190</td>
<td>0.500</td>
<td>6.636</td>
<td>0.052</td>
</tr>
<tr>
<td>1 MS/s</td>
<td>12.280</td>
<td>0.450</td>
<td>1.000</td>
<td>12.386</td>
<td>0.106</td>
</tr>
<tr>
<td>500kS/s</td>
<td>24.020</td>
<td>0.540</td>
<td>2.000</td>
<td>23.887</td>
<td>0.133</td>
</tr>
</tbody>
</table>

Figure 8. I and Q signals as captured by the USRP. The edge of the digital signal $S_m$ shows the start of transmission instant.

E. Sinusoidal carrier modulated by a commercial transceiver

The experiment of this section is aimed to evaluate the timing performance of the proposed system when a commercial transceiver is used to transmit the sinusoidal carrier, as shown in Fig. 9. In addition to the WUSS, there is a commercial transceiver (Freescale MC13192) inside a SARD (Sensor Application Reference Design) evaluation board that is used as a radio frequency source. The trigger signal $S_m$ is connected to the interrupt line of the SARD microprocessor (Freescale HC08) which has been programmed
to disable the attenuator in the transceiver and to transmit the carrier for 1 ms in correspondence of the interrupt.

Since the trigger frequency is still 250 Hz with duty cycle of 25%, the WUSS is in the same conditions of the experiment described in the previous section.

![Experimental setup for the evaluation of front-end latency and jitter.](image)

The results of the experiment are shown in Table III varying the sample rate and taking 1000 measures for each run. Whereas the internal processing delay $N_{\text{delay}}$ of the FPGA (expressed in samples) remains unvaried, the SARD exhibits a very high latency from the instant the interrupt is received to the time the attenuator is disabled in the transceiver. This behavior depends on the serial communication between the microprocessor and the transceiver. Moreover, this delay is not strictly constant and a calibration is needed for each transceiver type (not for every sensor!). A least square fitting is applied to the data in order to obtain the expected delay value of Table III. The absolute error of the average value with respect to the expected value benefits of the calibration procedure and it is lower (but in the same range) that the one reported in the previous section. The use of a commercial transceiver as sinusoidal carrier generator does not affect the time measuring performance of the proposed system.
TABLE III. MEASURED DELAY (μS) IN THE CASE OF A SINUSOIDAL CARRIER MODULATED BY A COMMERCIAL TRANSCEIVER.

<table>
<thead>
<tr>
<th>Sample rate</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Jitter</th>
<th>Exp.</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 MS/s</td>
<td>113.040</td>
<td>0.059</td>
<td>0.375</td>
<td>113.035</td>
<td>0.005</td>
</tr>
<tr>
<td>4 MS/s</td>
<td>114.400</td>
<td>0.123</td>
<td>0.250</td>
<td>114.406</td>
<td>0.006</td>
</tr>
<tr>
<td>2 MS/s</td>
<td>117.150</td>
<td>0.230</td>
<td>0.500</td>
<td>117.148</td>
<td>0.002</td>
</tr>
<tr>
<td>1 MS/s</td>
<td>122.630</td>
<td>0.480</td>
<td>1.000</td>
<td>122.631</td>
<td>0.001</td>
</tr>
<tr>
<td>500kS/s</td>
<td>133.600</td>
<td>0.800</td>
<td>2.000</td>
<td>133.601</td>
<td>0.001</td>
</tr>
</tbody>
</table>

F. Data packet transmitted by a commercial transceiver

In the previous experiments the delay measurements have been done using a sinusoidal carrier (as RF signal). Applying such a condition to a real application could be difficult, thus in this section the case of a delay measurement using a standard data packet is investigated. The experimental setup is the same of Fig.9, but the SARD evaluation board has been programmed to transmit a packet lasting for 1.1ms, after an external interrupt. The packet is composed of 32 bytes (256 bit) and is coded according to IEEE802.15.4 specification. As usual, the signal $S_m$ triggers a packet transmission every 4 ms. The signals acquired by a USRP on I and Q channels are shown in the time domain in Fig. 10 when sampling rate is set to 8 MS/s. It should be noted that the captured signal can be correctly decoded by an IEEE802.15.4 demodulator only if the sampling rate is higher than 4 MS/s; lower sampling rates allow for delay measuring only.

The experimental results are shown in Table IV. As in the previous experiment, the SARD evaluation board introduces latency due to the serial data transfer, since the packet transmission starts just after the
whole data are transferred from the microprocessor to the transceiver. For this reason, a least square calibration is applied to the measures to calculate the expected delay value.

**TABLE IV. MEASURED DELAY (µS) IN THE CASE OF A DATA PACKET TRANSMITTED BY A COMMERCIAL TRANSCEIVER.**

<table>
<thead>
<tr>
<th>Sample rate</th>
<th>Ave.</th>
<th>Std. Dev.</th>
<th>Jitter</th>
<th>Exp.</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 MS/s</td>
<td>1025.165</td>
<td>0.335</td>
<td>2.375</td>
<td>1025.625</td>
<td>0.460</td>
</tr>
<tr>
<td>4 MS/s</td>
<td>1027.378</td>
<td>0.620</td>
<td>3.500</td>
<td>1027.032</td>
<td>0.346</td>
</tr>
<tr>
<td>2 MS/s</td>
<td>1030.100</td>
<td>0.610</td>
<td>4.500</td>
<td>1029.843</td>
<td>0.256</td>
</tr>
<tr>
<td>1 MS/s</td>
<td>1035.340</td>
<td>0.780</td>
<td>4.000</td>
<td>1035.466</td>
<td>0.126</td>
</tr>
<tr>
<td>500kS/s</td>
<td>1046.780</td>
<td>1.020</td>
<td>8.000</td>
<td>1046.715</td>
<td>0.015</td>
</tr>
</tbody>
</table>

![Figure 10. I and Q signals as captured by the USRP when the SARD transmits a IEEE802.15.4 packet.](image)

The results show that there is a high residual error even after the calibration. This error is due to the uncertainty of the transmission instant that is internally decided by the transceiver; it cannot be measured nor
compensated. Thus, if a standard data packet is used instead of a sinusoidal carrier, the use of this type of transceiver greatly affects the time measuring performance of the proposed system. However, a different transceiver model with a less uncertainty on the transmission instants could be more effective.

VI. CONCLUSION

Many wireless sensors use monolithic transceivers for implementing the radio communication. Such an approach optimizes costs but has a very low flexibility; a device cannot react to changes in the surrounding environment.

This paper describes a new architecture, the WUSS, for smart sensors based on Software Defined Radio paradigm, which is a flexible and versatile hardware that adjusts its behavior by means of software reprogramming. The first benefit is that the WUSS can adapt its communication protocol and modulation to the environment and uses multiple channels and multiple standards for redundancy. The second benefit is that the WUSS can share the same radio hardware interface between the radio communication path and the sensor related interface. The WUSS is primarily designed to operate as microwaves-based measuring system, where the microwaves are used to make measurements like time-of-flight or localization. The use of a single, reprogrammable, RF interface boosts the versatility of the system and can reduce costs.

In order to demonstrate the feasibility of the proposed architecture, a prototype has been realized and characterized in this work. The experiments show that the WUSS is able to manage two independent and concurrent communications with the same RF front-end, transmitting on request the measurements taken from any standard sensor. Besides, the WUSS prototypes based on a USRP motherboard can achieve delay and phase measurements with a very low standard deviation (3 ns) and hence they can be effectively used, for instance, for time-of-flight measurement. Last, the WUSS prototype, which operates without modifying the hardware front-end of a standard SDR platform, exhibits the ability to estimate the timestamp of generic
RF events with error of about 20 ns. Such a feature allows for time synchronization among different nodes of the same network and it could also help in the localization of the nodes.

ACKNOWLEDGMENTS

The authors would like to thank Ing. C. M. De Dominicis for her valuable help given to the experimental work.

REFERENCES


[10] Li, Yafei and He, Zhitao and Voigt, Thiemo and Leidelöf, Sanna, “A software radio-empowered sensor network,” 9th Scandinavian Workshop on Wireless Adhoc Networks (Adhoc'09), 4-5 May 2009, Uppsala, Sweden


