1st International Conference on Sensing Technology
November 21-23, 2005 Palmerston North, New Zealand

A Low-cost Distributed Measurement System based on Gas Smart Sensors for Environmental Monitoring

L. Bissi(*), A. Scorzoni(*), P. Placidi(*), L. Marrocchi(*), M. Bennati (*),
S. Zampolli(**), L. Masini(**), I. Elmi(**) and G.C. Cardinali(**)
(*)Dipartimento di Ingegneria Elettronica e dell'Informazione, University of Perugia, via G.Duranti 93, 06125 Perugia, Italy, lucia.bissi@diei.unipg.it
(**)CNR – IMM, Sezione di Bologna, via Gobetti 101, 40129 Bologna, Italy

Abstract

In this paper we describe the structure of a multiprocessor control system for a gas sensing array, inspired to the IEEE 1451 standard. The system (Smart Transducer Interface Module, STIM) features a simplified Transducer Independent Interface (TII) based on a 3-wire RS232 communication and is conceived as a cluster between identical monosensor subsystems and a central Controller. After a brief illustration of the modular system architecture, an overview on the basic monosensor modules and of the simplified TII will be given. Then, the design and implementation of a gateway acting as an interface between the system and the Ethernet (Network Capable Application Processor, NCAP) will be described. Finally, experimental results will be presented.

Keywords: smart sensors, environmental monitoring, metal-oxide sensors, IEEE – 1451 Standard

1 Introduction

Detection of gases or vapors in air is becoming important mainly in the context of environment control and safety. Therefore a great effort is applied to realize low-cost, compact instruments that can detect the presence of chemical compounds and pollutants. Embedded with a microcontroller unit or a microprocessor, a smart sensor has much more built-in intelligence over a traditional sensor [1 – 3]. Therefore it can perform more powerful functions such as a self-identification, and a self-calibration. However, as to the distributed measurements applications [4], one of the most attractive advantages that a smart sensor offers is the networking capability regulated by the IEEE 1451 smart transducer interface standards. The aim of this standard, which is composed of four parts [5 – 8] plus a proposal reported in ref. [9], is to enable plug and play transducer level, standardizing data structures and communication, and to simplify the creation of networked sensor model over a network independent system. However, the IEEE 1451 standard specifies at least two layers between the sensor/actuator and the network: first, the Smart Transducer Interface Module (STIM) to interface sensors and hold their TEDS data (Transducer Electronic Data Sheet, [6]), second, the Network Capable Application Processor (NCAP, [5]) that supports network communications and it is able to run programs locally and to cooperate with other NCAPs through a network.

A few prototypes of NCAPs and STIMs have been proposed in the past. The EDI520 CogniSense module from Electronics Development Corporation (EDC) combining a PIC and signal-conditioning ASIC could be programmed to function as a 1451.2 STIM. A small piezoresistive accelerometer manufactured by IC Sensors was directly connected to the EDI520. A development kit was produced by EDC to get started with 1451.2. It included an NCAP and two STIMs as well as software. The NCAP interfaced with a RS485 network. The company Esensors produced a "websensor" with a 1451.2 prototyping kit composed by a EM04a NCAP (with ES00r for configuration) and a EI02 STIM with cables. National Instruments advertised a 1451 interface on Labview. Finally, Hewlett-Packard produced an Embedded web server (HP BFOOT 66501) that was soon discontinued.

The standard connection between the STIM and the NCAP layers is the so-called Transducer Independent Interface (TII), a 10 wire bus designed for feeding the STIM and for exchanging information [6]. A key reason for standardizing the interface at the hardware interconnection level is due to the compatibility problems transducer manufacturers face when integrating their devices into multi-vendor networks [4]. Transducer interfacing also requires standardized software interfaces to provide application and network interoperability at the network-node level. Because the network and the transducer must expose their interfaces directly to transducer applications on each node, any attempt to migrate the application, the sensor hardware, or the network node to another platform requires a time-consuming and costly redesign of the application's interface to the environment. If standard interfaces such as the IEEE 1451 are used, then sensor – to –sensor interchangeability and sensor – to – network interoperability can be realized.

Since most of the commercial systems compliant to the IEEE 1451 standard have been discontinued, an unpleasant conclusion that we draw is that, these products were possibly limited by the TII standardized interface which, due to its inherent complexity, acted as a bottleneck instead of being helpful for a widespread use of the standard.

In this paper we describe a modular system designed for detecting ambient pollutant species through an array of gas sensors. The system relies on a modular microcontroller architecture, suitable for low cost applications. The management of the system basically follows the IEEE 1451 directives. However, the TII has been replaced with a more conventional and widely used 3-wire RS232 serial interface. It should be noted that a substitution of the TII with a RS232 interface was previously proposed [10]. However, in that case a full 9-wire interface was adopted.

The organization of the paper is as follows. First, an overview of the STIM modular architecture and its features is given. In particular, the monosensor subsystem and the interface between STIM and NCAP will be illustrated. Then, an NCAP based on simple hardware and capable of interfacing the STIM with the Ethernet will be described. Finally we will report the on measurements performed in a controlled environment.

2 The proposed STIM

The STIM system controls and acquires data from a gas sensor which exploits an array of four metal-oxide thin films deposited on thin dielectric membranes [11, 12]. A multisensor system is preferred because a single sensor does not react selectively to a singular target gas inside a mixture, but also to other interfering gases, therefore sensor redundancy helps in improving the performance of the gas detector. The system is based on a microcontroller local network, an advantage of this architecture is the modularity of the system.

Figure 1 shows the proposed system architecture. The STIM, as defined in the IEEE 1451.2 [6] directive, has been conceived by exploiting a modular approach: a central Controller is connected via I^2C (Inter-IC bus) interface to four Satellites.

To simplify the communication between the STIM and the NCAP the RS232 serial interface has been used to emulate the complex Transducer Independent Interface (TII), defined in the IEEE 1451.2 standard.

In the first version of the system, instead of building a dedicated NCAP device, an NCAP module emulator is realized through a LabView Development System™ Virtual Instruments (VI) on a personal computer (softNCAP) featuring an RS232 serial interface communication. It presently allows the user to control and monitor the sensors system through a local PC. Section 3 describes how the system could be connected to the Internet with the Ethernet protocol by exploiting a "hard" NCAP.

In the communication, the softNCAP is the master and the STIM is the slave, this protocol is based on reading and writing memory registers that store the information on the status of the system and of the performed operation. The 1451 softNCAP addressing operation is accomplished by sending the STIM a 2 byte address containing the functional address, which indicates the type of operation, and the channel address to specify the number of the receiving channel.

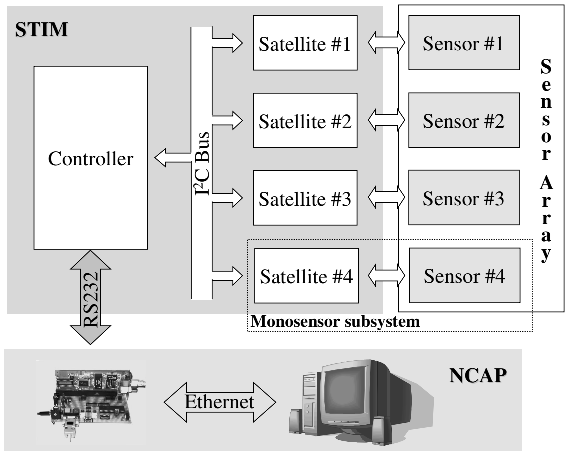

Figure 1: Block diagram of the system.

The Controller is a Microchip PIC16F876A microcontroller (μC). It receives the addressing requests from the NCAP through the built-in USART hardware module. The μC decodes the commands and, when applicable, echoes them in a simplified form to the Satellites through the I^2C bus. After the execution of the operation, the Satellite sends an acknowledge back to the Controller.

The Transducer Electronic Data Sheet (TEDS) is stored in the permanent memory and it contains the relevant data of the sensors. In the system, the Meta-TEDS and the Channel TEDS are stored in the Flash memory of the Controller, while the operating parameters of each sensor are stored in the Generic Extension TEDS located in the EEPROM of the relevant μC that manages the Satellite (a Microchip PIC16F873). Of course, the softNCAP allows the user to set the Generic Extension TEDS of each sensors.

Following the IEEE 1451.2 directives, the channels of our system are defined as Buffered Data Sequence Sensor [6], i.e. the sensor acquires data continuously, with sampling times under control of the STIM. A stack is continuously rewritten when acquiring new measurement data and a pointer is updated. The stack is copied in a dedicated buffer when the trigger is sent by the softNCAP. Then this buffer is available for reading through the modified TII. The stack and the buffer are placed in the RAM of the Satellite μC. The memory management of the μCs used in this architecture allow for managing up to twelve independent channels using a modular approach.

2.1 Monosensor subsystem

Each monosensor subsystem (shown in Figure 2) [1] is composed of a sensor and a Satellite.

A Satellite can be further divided into the following blocks:

- a μC featuring I^2C interface;
- a Temperature Control Circuit for the sensor;
- a DAC block which provides a reference voltage (V_{ref}) for the control of the operating temperature T_{op};
- a Conditioning Circuit for translating the resistance of the sensing layer into a voltage suitable for the A/D conversion of the μC.

The I^2C interface, managed by the μC, allows the Satellite to communicate with the Controller. The Controller sends the Satellites the commands and the operating parameters and they reply accordingly.

The Temperature Control Circuit is in charge of maintaining a uniform temperature T_{op} of the sensing layer. The Heater is characterized by a calibration curve between the heater resistance (R_H) and T_{op}. So the T_{op} can be kept constant by comparing the value of R_H (or, equivalently, the voltage V_H) with a reference resistance R_{ref} (or, equivalently, a V_{ref}) which corresponds to the chosen T_{op} in the calibration curve.

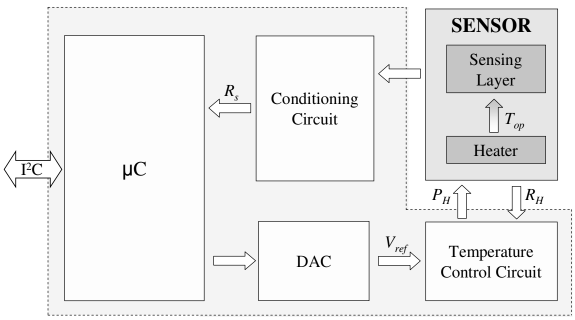

Figure 2: Block diagram of a monosensor subsystem.

In the Satellite the output of a PWM module of the μC is filtered by a low pass filter in order to emulate a DAC, providing the V_{ref} voltage to the Temperature Control Circuit.

The resistance of the Sensing Layer (R_s) is measured with a potentiometric measurement. Suitable analog components are adopted in the Conditioning Circuit for the pre-elaboration of the acquired signal before the A/D conversion of the μC and the subsequent calculation of the R_s value. A wide dynamic measurement range (5kΩ, 15MΩ) of R_s was obtained using two ranging resistors. The appropriate ranging resistor is chosen through an auto-range procedure managed by the μC.

Two kinds of temperature control are provided: “continuous” when the temperature T_{op} of the sensing layer is kept constant, “pulsed” when heating periods of the Sensing Layer (at temperature T_{op}) are interleaved with cooling periods at ambient temperature (T_{amb}). The choice of the temperature control depends on the particular sensor material and on the particular type of gas to be detected.

The R_s value, after appropriate elaborations, provides the concentration of a particular kind of gas we want to detect.

The electrical behavior of the system is influenced by device tolerances and variation of parameters with respect to nominal values. In order to compensate for possible variations of the parameters, an automatic self-calibration procedure has been implemented, managed by the μC firmware.

2.2 Redesigned interface between STIM and NCAP

In the IEEE 1451.2 standard [6] the interface between the STIM and softNCAP is the Transducer Independent Interface (TII). This interface presents a complex hardware structure, featuring ten physical lines belonging to four groups: signals for Data Transport (DATA_OUT, DATA_IN, DATA_CLOCK, N_IO_ENABLE), for Triggering (N_TRIGGER), Support (POWER, COMMON, N_ACKNOWLEDGE, N_STIM_DETECT) and Interrupt (N_IO_INTERRUPT) lines. The signals carry binary data or control that generally are active low, therefore their names are prefixed with an N as a mnemonic aid.

In our system a simple 3-wire RS232 serial interface is used to implement the complex TII. The hardware signals have been replaced by simple messages running on the RS232 interface and the 1451.2 communication protocols are maintained. Each signal of the TII is mapped with a byte that is transmitted through the RX-TX lines of RS232, and so the parallel communication is implemented on a serial interface. Without the clock signal of the TII (DATA_CLOCK), the synchronism is maintained with the NCCOMM message. Table 1 shows the correspondence between TII signals and RS232 equivalent messages. Since hardware signals can be asserted or negated, it was necessary to define a byte message for each signal level.

In the communication protocol the Triggering and Data Transport operations have been implemented as separate tasks, being mutually exclusive.

The triggering operation is normally used before reading a sensor. During the quiescent state of the STIM, the softNCAP sends the NTRIG_ASSERTED message. The STIM sends the triggering command to the triggered channel and the relevant Satellite copies the values of the stack on the trigger buffer and updates the Status Register. Then the STIM asserts the trigger acknowledgment signal sending the NACK message to the softNCAP that replies with a NTRIG_NEGATED message. The STIM ends the communication by sending a NACK message.

The NACK message sets the data flow between the softNCAP and STIM: when the STIM receives or sends a data byte, it sends the NACK message to the softNCAP. This message means that the STIM can manage other data. The softNCAP is the master of the communication, it controls the data transfer with the NCCOMM message, that is sent after receiving or transmitting the data byte to communicate to the STIM the continuation of the transmission.

Table 1: Comparison between TII lines and RS232 equivalent signals.

TII lines	RS232 equivalent messages
DATA_OUT (DOUT)	STIM TX – NCAP RX
DATA_IN (DIN)	NCAP TX – STIM RX
DATA_CLOCK (DCLK)	—
—	NCCOMM 'C'
N_IO_ENABLE (NIOE)	NIOE_ASSERTED 'I' NIOE_NEGATED 'i'
N_TRIGGER (NTRIG)	NTRIG_ASSERTED 'T' NTRIG_NEGATED 't'
POWER	—
COMMON	—
N_ACKNOWLEDGE (NACK)	NACK 'A'
N_STIM_DETECT (NSDET)	NSDET 'S'
N_IO_INTERRUPT (NINT)	NINT_ASSERTED 'Q' NINT_NEGATED 'q'

The handshaking protocol of Data Transport is shown in Figure 3. The NCAP sends the NIOE_ASSERTED message to the STIM that replies with the NACK message. Then the NCAP sends the functional address byte to ask for an operation and, after receiving the NACK message from the STIM, it sends the channel address byte to select the channel. The NCAP can receive or send a sequence of data byte (each of them followed by the NACK message). At the end of data transfer, the NCAP sends the NIOE_NEGATED message and the STIM replies whit the NACK message.

3 Interfacing with Ethernet

In order to achieve the maximum flexibility of the system, using the same protocol and software, we implemented an embedded gateway between the modified RS232 TII standard and the TCP/IP network over Ethernet (Figure 4). This smart system operates like an NCAP, being capable to interconnect the STIM module, as described in the previous sections, over the well-know and widely used Ethernet network.

There is nothing new implementing smart sensor interfaces using the TCP/IP suite of protocols and the powerful interface given by the HTML protocol and programming language, as can be seen in refs. [13,14], but the system proposed here is fully transparent and independent of software and hardware used on both sides of connection. This gives more flexibility to the overall system yielding a practical independence of the standardization procedures on both sides of the connection (Ethernet and RS232).

While the TCP/IP stack is a simple task for a PC-based machine, capable of thousand million operation and featuring a large amount of memory, the same suite of protocols is not as easy to implement in a limited-memory machine like today's microcontrollers.

Using efficient coding and optimized compilers, today microcontrollers are capable to handle a reduced version of the TCP/IP protocols giving system developers the possibility of producing cost-effective devices with added networking capabilities.

In this particular application the core of the system is a microcontroller produced by Microchip and belonging to the PIC18 series [15]. Table 2 shows the main characteristics and performances.

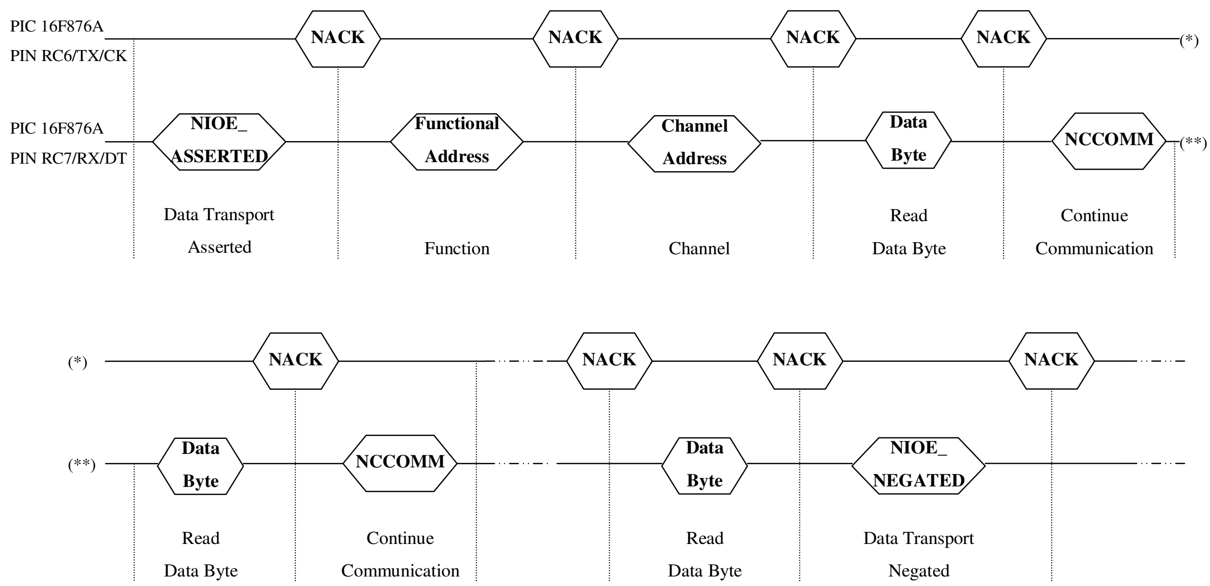

Figure 3: Example of Data Transport operation between STIM and NCAP.

Compared to typical memory consumption of a TCP/IP stack in a PC [16], the figures reported in Table 2 lead us to the conclusion that the resources and memory of the microcontroller should be carefully optimized in order to get a reliable TCP/IP communication.

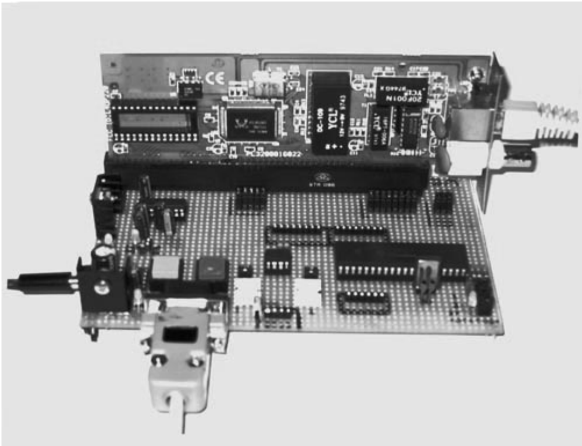

Figure 4 : Realization of the TCP/IP to RS232 embedded gateway.

The implemented system, acts like a transparent gateway for accessing the STIM module through the Ethernet network rather than through the serial connection previously described.

Table 2: Main performances of the chosen microcontroller.

Hardware	8 bit RISC MPU
Flash ROM Memory	16 kword (32 kbyte)
Internal SRAM	1 kbyte
Performance	10 MIPS (max)

Figure 5 shows the data flow between the Ethernet network and the RS232 serial protocol. The application starts setting up the USART registers of the microcontroller, in order to comply with the RS232 communication standard. Incoming and outgoing data, from and to the Ethernet network are managed by the Ethernet front-end, realized using a Realtek RTL8019AS chip [17], and are transmitted to/from the microcontroller using two ring buffers. Subsequently the TCP/IP stack [18] reliably manages the connection and the data stream between the interested nodes. Data sent to the Ethernet network reside in a ring buffer, which avoids data loss and improves the efficiency of elaboration of the main application; instead, received data are elaborated on a byte basis, since the recovery procedure has to be fast in computational terms.

In order for the gateway to exploit the available softNCAP application in LabView, it was necessary to redirect on the PC the RS232 connection over the Ethernet network on TCP/IP protocol stack [19]. Using this methodology the connection between the sensor network (STIM) and the Ethernet becomes completely transparent and compliant to the IEEE 1451 standard, allowing users to directly exploit the previously developed system.

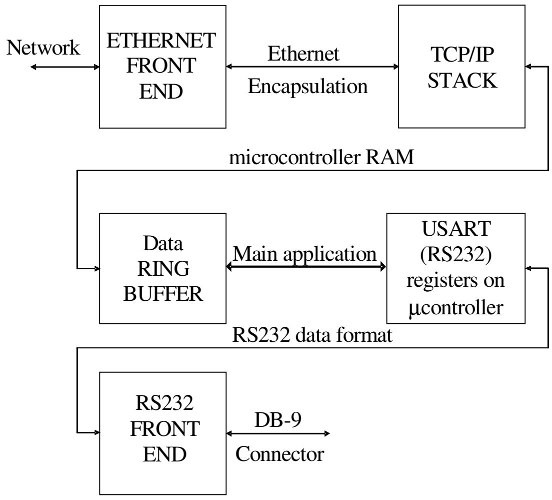

Figure 5: Logic block diagram of the system.

4 Results

After careful debugging, the system has been tested by acquiring data of a metal-oxide gas sensor array located in a suitable environmental chamber under a flow of 500 sccm. In Figure 6, some typical sensor responses to environmental pollutants are reported. The upper plot shows the resistance of a SnO_2 thin film gas sensor with a thin layer of gold catalyst, while operated at a constant temperature of 400°C. The lower plot shows the sensor resistance before (pre-pulse), during and after (post-pulse) a 100 ms temperature pulse at 375 °C.

The first response (at t = 4.6 hours) is relative to an injection of a mixture of Benzene, Toluene and m-Xylene, each at a concentration of 0.5 ppm for 10 minutes and 1.0 ppm for the following 10 minutes. The second pattern (at t = 6.4 hours) is the response to 25 and 50 ppm of carbon monoxide, while the last pattern (at t = 8.1 hours) is the response to 0.25 and 0.5 ppm of nitrogen dioxide.

As can be disclosed from these plots, the sensor operated at constant temperature has a good response to the aromatic compounds, while the sensor in pulsed mode is selective towards nitrogen dioxide, although the sensor response time is slightly longer than in constant temperature mode.

5 Conclusions

This paper describes a multiprocessor control system based on a microcontroller local network that detects ambient pollutant species through metal-oxide chemical gas sensors.

Thanks to its modularity, the system can be expanded including additional monosensor subsystems while keeping the same architecture.

The system, realized following the IEEE 1451 directives, is presently composed of a Controller and four Satellites. Each Satellite acquires resistance values from a single gas sensor.

After debugging, the system has been tested in environmental chambers under a known gas sequence. Promising results were obtained.

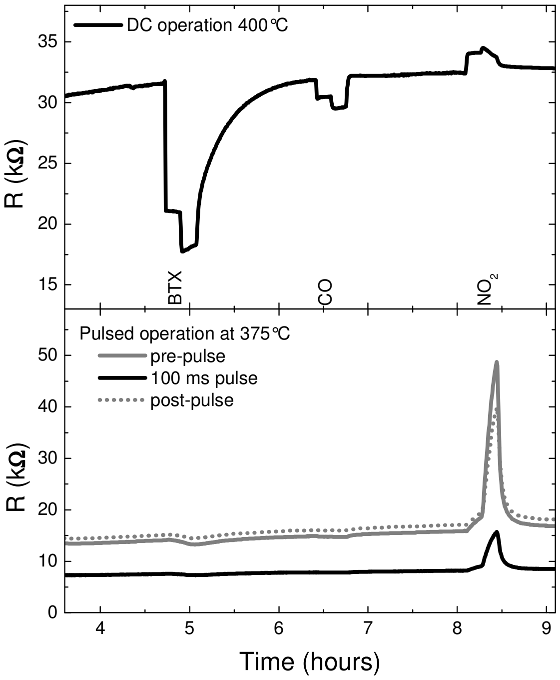

Figure 6: Sensor response to environmental pollutants.

6 Acknowledgements

The partial financial support of the FISR-MIUR Project "Microsistemi per applicazioni ambientali e agroalimentari" is gratefully acknowledged.

7 References

[1] Baroncini, M., Placidi, P., Cardinali, G.C., Scorzoni, A., "A simple interface circuit for micromachined gas sensors", *Sensors and Actuators A: Physical,* Vol 109/1, pp. 131-136, (2003).

[2] Depari, A., Ferrari, P., Flammini, A., Marioli, D., Sisinni, E., Taroni, A., "IEEE1451 smart sensors supporting USB connectivity", *Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE*, pp.177 – 182, (2004).

[3] Ferrari, P., Flammini, A., Marioli, D., Taroni, A. "A low-cost Internet-enabled smart sensor", *Proceedings of IEEE Sensors*, 2002, Volume 2, 12-14 June 2002, pp. 1549 – 1554, vol.2, (2002).

[4] Lee, K.B., Schneeman, R.D., "Distributed measurement and control based on the IEEE 1451 smart transducer interface standards", *IEEE Transactions on Instrumentation and Measurement,* Volume 49, Issue 3, June 2000, pp. 621 – 627, (2000).

[5] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators: Network-Capable Application Processor (NCAP) Information Model", *IEEE Standard 1451.1-1999*, (1999).

[6] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators: Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats", *IEEE Standard 1451.2-1997*, (1997).

[7] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators: Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems", *IEEE Standard 1451.3-2003*, (2003).

[8] "IEEE Standard for A Smart Transducer Interface for Sensors and Actuators: Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats", *IEEE Std 1451.4-2004*, (2004).

[9] IEEE P1451.5 Project, "Draft Standard for A Smart Transducer Interface for Sensors and Actuators - Wireless Communication Protocols and Transducer Electronic Data Sheets (TEDS) Formats",http://grouper.ieee.org/groups/1451/5/.

[10] Johnson, R.N., "Serial Implementation of IEEE 1451.2-1997 for Data Acquisition Applications", Sensors expo Chicago, June 3, 2002.

[11] Scorzoni, A., Baroncini, M., and Placidi, P., "On the relationship between the temperature coefficient of resistance and the thermal conductance of integrated metal resistors", *Sensors and Actuators A: Physical*, Vol. 116, Issue 1, pp. 137-144, 4 October 2004, (2004).

[12] M. Baroncini, P. Placidi, G.C. Cardinali and A. Scorzoni, "Thermal characterization of a microheater for micromachined gas sensors", *Sensors and Actuators A: Physical*, Volume 115, Issue 1, pp 8-14, 15 September 2004, (2004).

[13] Flammini, A., Ferrari, P., Sisinni, E., Marioli, D., and Taroni, A., "Sensor integration in industrial environment: from fieldbus to web sensors", *Computer Standards & Interfaces,* p. 2173 (2002).

[14] Flammini, A., Ferrari, P., Sisinni, E., Marioli, D., and Taroni, A., "Sensor interfaces: from field-bus to Ethernet and Internet", *Sensors and Actuators A*, Vol. 101, pp.194–202 (2002).

[15] Microchip Technologies, "*PIC18F452 Datasheet*", http://www.microchip.com.

[16] Bentham, J., "*TCP/IP Lean*" *Second Edition*, Lawrence – Kansas (USA), CMP Books

[17] Realtek Semiconductor, "*RTL8019AS Specification*", http://www.realtek-tw.com.

[18] Microchip Technologies, "*The Microchip TCP/IP Stack*", Application Note 833, 2002, http://www.microchip.com.

[19] TALtech, "*TcpCom - RS-232 to TCP/IP software converter*", http://www.taltech.com.