Teaching Communication Protocols Used In Instrumentation Laboratory

C.G. Haba, L.Breniuc

"Gh.Asachi" Technical University of Iasi, Bd.D.Mangeron Nr.53, 700050 Iasi, Romania
E-mail: cghaba@ee.tuiasi.ro, lbreniuc@ee.tuiasi.ro

Abstract-We present a system for teaching different serial communication protocols used in instrumentation laboratory. The system is implemented using XESS prototyping boards (XSBoards) each one equipped with either a Xilinx FPGA or CPLD and an 8031 microcontroller (μC)[1] and several intelligent sensor types. The system was used for implementing and testing the I^{2}C, SPI, 1-Wire and other company specific protocols. The communication protocols were tested by implementing measuring systems where a sensor or a network of sensors is configured and controlled by a microcontroller.

I. Introduction

In the last decade there was an important development in the measurement instruments technology. Connecting different sensors and instruments in a distributed environment such as a laboratory or a factory is usually achieved using conventional parallel and serial connections. However, by far the most important networking technique in a distributed environment beyond the size of a test bench is the serial bus.

The Inter Integrated Circuit Bus (I^{2}C) bus is a bidirectional two-wire serial bus introduced by Philips and widely used in the industry for embedded applications. A large set of devices can be connected to the I^{2}C bus such as: sensors, microcontrollers, RAM and EEPROM memories, LCD displays, ADCs, real-time clocks etc.

The Serial Peripheral Interface was introduced by Motorola in their microcontrollers and it is a bidirectional full-duplex four-wire serial bus.

A simplified design that lowers system cost can be achieved using 1-Wire bus introduced by Dallas Semiconductors. It is a serial bus that supplies control, signalling and power over a single wire connection. Data transfer is half duplex between a master and more slave devices and it is initiated and controlled by the master.

In our paper we present a system used in the digital instrumentation laboratory for teaching communication protocols operating over different serial communication systems.

II. Equipment and software

The central focus of the laboratory is to teach digital measurements and measurement microsystems and it has the necessary equipment to support multiple sets of student experiments.

For our set of tools, we have allocated three PCs, three XS40-010XL(E) boards. There are also available several intelligent sensors and components with serial communication capabilities.

The XS40-010XL(E) board has as main components a Xilinx FPGA programmable circuit XC4010XL (XC4010E) and an 8031 microcontroller. It also contains 128 Kbytes SRAM, 100 MHz programmable oscillator, parallel port, mouse/keyboard PS/2 port, VGA monitor port, 7-segment LED, 84-pin breadboard interface, serial EEPROM socket, 9V DC power jack and 5V/3.3V regulators.

The board requires external 9VDC power supply and can be connected to the PC via the parallel port. Using the available ports, we can connect to the board a VGA monitor, a PS/2 mouse or a keyboard.

Several software tools are required for the successful completion of the laboratory works. These tools are needed for:

- designing the hardware part to be implemented in the programmable circuit;
- writing and debugging software part (microcontroller programs in C or assembler language);
- configuration of the onboard programmable circuit;
- loading microcontroller program into memory;
- testing the design by sending test vectors to the board.

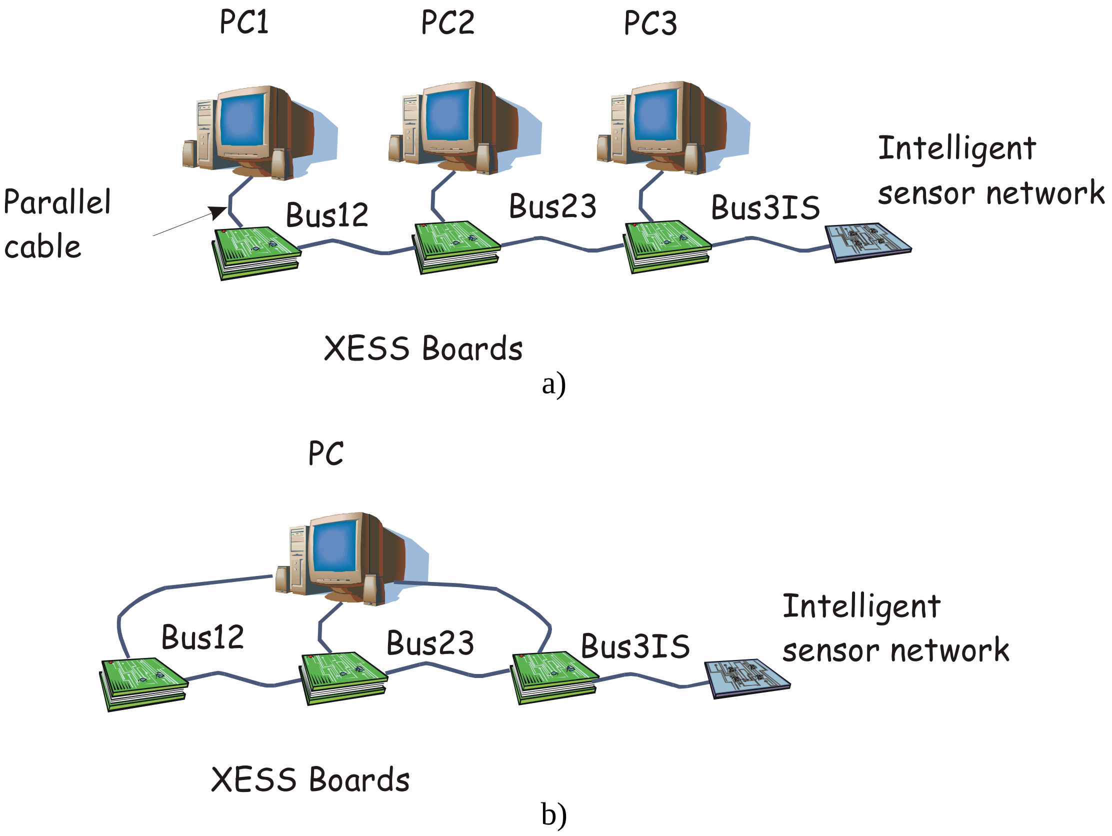
Software installed on the computers to be used with the system includes: Xilinx Foundation [2]; Xess XSTools [3]; Franklin Software for programming the 8031/51 μC including also a C compiler.

XESS XSTools consists of additional software for connecting the XS boards to the PC parallel port and consists of four components:

- *xsload* for downloading the configuration files to the FPGA and the program file for the μC;
- *xsport* for applying signals using the parallel port;
- *xstest* for testing if the boards are well functioning;
- *xssetclk* for setting the frequency of the programmable oscillator.

III. Implementation

The system consists of three XSBoards, each one connected to the parallel port of one of the personal computer via a parallel cable (Fig. 1a). Using a parallel port extension card the system can be operated using only one PC (Fig. 1b).



a)

b)

Figure 1. System configuration: a) multiple PC configuration; b) single PC configuration.

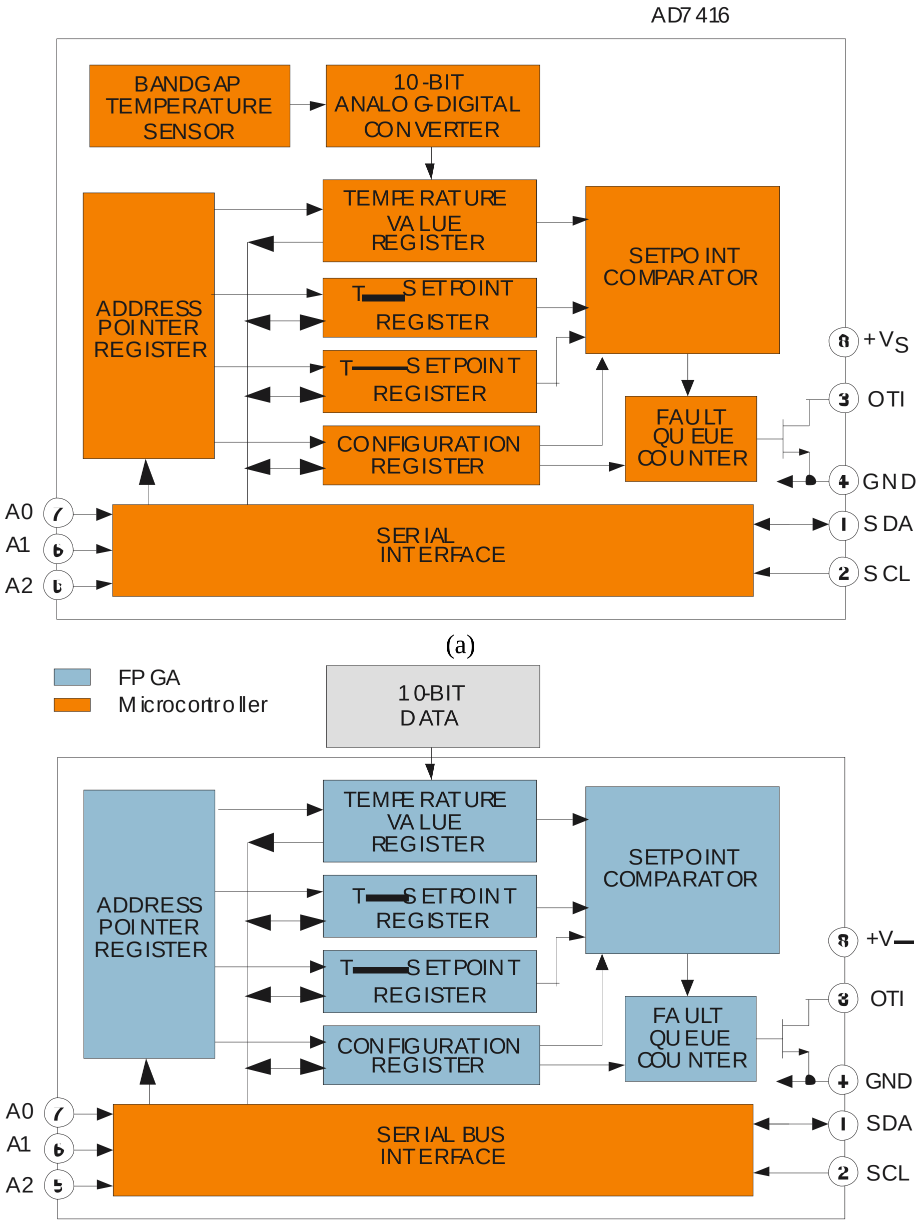
Part of the pins of the XSBoards (corresponding to pins of the FPGA) are connected using serial cables and form two buses, Bus12 and Bus23 respectively. By programming the FPGA on the middle XSBoard, pins of XSBoard1 can be routed to pins of XSBoard2 resulting in different connection configurations. A third bus, Bus3IS connects the third XSBoard with a fourth board which hosts a network of intelligent sensors. The network of sensors is used to test the communication protocols implemented using the XSBoards.

Based on this system there can be set several lab works falling into following categories:

- implementation of the communication protocols;
- verification of the communication protocols;
- simulation of intelligent sensors.

A. Implementation of the communication protocols

The following communication protocols have been implemented using the 8031 microcontroller: I2C, SPI, 1-Wire and the specific protocol for the Sensirion humidity sensors. The first part of the laboratory work is dedicated to the study of the protocol in consideration. Flow-charts and time diagrams help students to understand the command sequences for the initiation and realization of the conversation between devices. The assembly or C code for the 8031 microcontroller that will implement the behaviour of a master or slave device is then analysed for the considered protocol. The protocol is subsequently tested running the code on the microcontrollers on the XSBoards. One microcontroller can be designated to be the master and the other two to be the slaves.



(a)

(b)

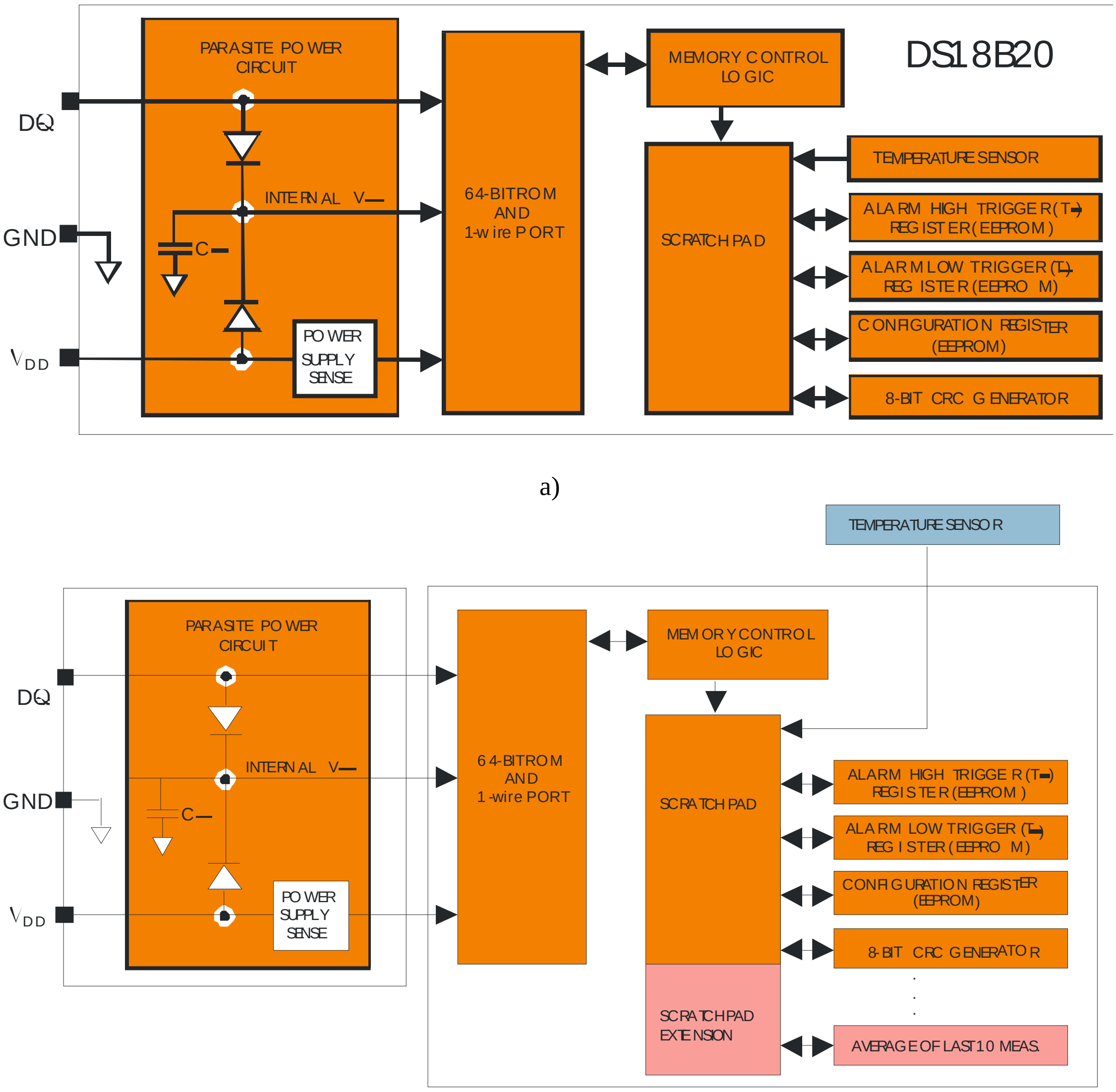
Figure 2. AD7416 sensor: a) original block diagram; b) simulated sensor block diagram.

B. Verification of the communication protocols

During laboratory works students have the possibility to verify the implemented communication protocols in an environment with devices having integrated the communication protocol. A temperature measurement system has been considered which can be used to verify each communication protocol. The system allows the realization of a network of four intelligent sensors capable of communicating using the same protocol. Using a single board, with appropriate sockets, only by replacing the sensors we can obtain different types of sensor networks. The following table indicates the sensor type used for each of the temperature measurement system:

Table 1. Temperature measurement system sensors

Protocol	Sensor type	Part number
I^2C	temperature	AD7416 (I^2C) [5]
SPI	temperature	MAX6662[6]
1-Wire	temperature	DS18B20 [7]
Sensirion specific	temperature and humidity	SHT11[4]



a)

b)

Figure 3. DS18B20 sensor: a) original block diagram; b) simulated sensor block diagram.

C. Simulation of intelligent sensors

Following the comprehension of the communication protocols and understanding the structure of intelligent sensors students have the possibility to learn how to make a sensor intelligent by appending processing and communication capabilities. The XSBoards contain both an FPGA and a microcontroller, therefore additional capabilities can be implemented either hardware or software.

For example, the AD7416 whose block diagram is depicted in Figure 2(a) is simulated by a design using the XSBoard and some additional external components. In this design part of the sensor structure is implemented in the FPGA and part in the microcontroller. The temperature sensing element is simulated using either a 10-bit bar switch or a 10-bit digital potentiometer. Once the structure and protocol understood, students can customize the design in order to obtain other sensor types.

In Fig.3 we present the simulation of the 1-Wire DS18B20 temperature sensor. Students can extend the functionality of the sensor by modifying different blocks of the sensor or adding new ones. For example, the new sensor depicted in Fig.3b has a new function that computes the average of the last ten measurements. This value can be read from the last byte of an extension to the scratch pad memory. For this byte, a non-volatile storage is also provided.

Connecting the PC to the Internet, and using interfaces developed by the authors [8], the system can be access and used online.

IV. Conclusions

A teaching system for communication protocols has been implemented. Using this system students gradually learn how different protocols work, how can these protocols be implemented and how sensors can be made intelligent by adding necessary hardware and software resources allowing them to be configured and to communicate with other measurement system components.

References

1. ***, "*XS40, XS95, XSA Board User Manual*", X Engineering Software Systems Corporation, 2001.
2. D. Van den Bout, "The Practical Xilinx Design Lab Book", Prentice Hall, 1999.
3. ***, "XSTOOLs V4.0 User Manual. GUI-Based and Command Line-Based XS Board Utilities", X Engineering Software Systems Corporation. 2001.
4. ***, SHT1x/SHT7x Humidity and Temperature Sensmitter, Sensirion AG, June 2002, http://www.sensirion.com
5. ***, 10-Bit Digital Temperature Sensor (AD7416) and Single/Four-Channel ADC (AD7417/7418) , Analog Device Inc, 2000, http://www.analog.com.
6. ***, 12-Bit + Sign Temperature Sensor with SPI-Compatible Serial Interface, Maxim Integrated Products, 2001, http://www.maxim-ic.com.
7. ***, DS18B20 Programmable Resolution 1-Wire Digital Thermometer, Dallas Semiconductor Maxim, 2002, http://www.maxim-ic.com.
8. L.Breniuc, C.G.Haba, Web-Based Monitoring and Control of a Temperature Measurement System, Proceedings of the 2nd on-line Workshop on Tools for Education in Measurement, Tampere , Finland, 1-15 June 2002, pp. 11-15.