# RS-485 BASED MEASUREMENT SYSTEM WITH SCPI COMMAND SET CONTROLLED BY HP-VEE APPLICATION

**Grzegorz Zareba, MScCS**
**University of Arizona**

## ABSTRACT

This article presents a measurement system based on the RS-485 interface. The presented system is an alternative solution for distributed measurement systems, which cannot be built using IEEE-488 interface due to distance limitation between elements of the system. The RS-485 interface is a base for communication between measurement instruments and uses a Master-Slave protocol to exchange data between them. One dedicated master device, usually a PC, controls all slave devices connected to the interface. To control measurement devices SCPI language is used. This solution simplifies communication between measurement devices and allows utilizing the HP-VEE environment to control any SCPI devices connected to the RS-485 network.

## KEY WORDS

RS-485 Interface, Bus Networks, SCPI Language, IEEE-488 Interface, HP-VEE Environment, ModBUS Protocol.

## INTRODUCTION

Many measurement systems are based on the IEEE-488 interface. These kinds of systems are expensive but easy to use because most of the measurement instruments have built in IEEE-488 interface or are prepared to install the IEEE-488 bus card. The main disadvantage of these systems is that the distance between elements working on the IEEE-488 bus is highly restricted. Many measurement systems are distributed systems, which means that the distance between instruments and other elements of the system e.g. sensors are usually too big to involve IEEE-488 based systems. This problem can be solved using special techniques, which are unfortunately very complicated. Another disadvantage of the IEEE-488 based systems is the necessity to equip each element of the IEEE-488 system with an IEEE-488 card, which is hard to accomplish in case of small sensors.

One of the solutions is to build a system, which has features of a homogenous system and functionality comparable with the IEEE-488 based measurement systems.

As an alternative solution for measurement systems, the system based on the RS-485 interface with bus network architecture can be proposed. The RS-485 interface allows for connecting up to 32 devices according to a multi-point structure of the physical layer. Number of connected devices can be extended using repeaters. Communication between network elements is maintained using the ModBus protocol. One of the devices has to be a "Master" device and the rest of them, are "Slaves". The Master device is usually a PC or any other device, which controls data exchange in the whole system. Slave devices are measurement instruments or sensors. The ModBus protocol is used to pass the SCPI (Standard Command for Programmable Instruments) commands from a Master device to the addressed Slave device. Each device working in the system has to have an RS-232 interface and should be capable of interpreting the SCPI commands.

Main element of the proposed system is a communication device, which serves as an interface between measurement devices, comprising of the SCPI based measurement system. The communication devices create the RS-485 based transmission medium controlled by the ModBus protocol.

The HP-VEE environment (Hewlett-Packard Visual Engineering Environment) is designed for building applications using IEEE-488 or RS-232 interface, to communicate with measurement instruments. The HP-VEE application sends and receives SCPI commands to/from external devices via three kinds of objects: Direct I/O, Component Driver and Instrument Panel. The Instrument Panel and Component Driver need a special driver, which describes the communication with an external device on the level of SCPI commands exchange. Direct I/O objects can send any string to the external device.

## RS-485 BASED BUS NETWORK

The RS-485 [1] interface allows building simple bus networks, which can be controlled by a master-slave protocol. A master-slave protocol is a deterministic protocol in which one device, master device, has special functions used to control an access to the common transmission medium. All other devices, slave devices, wait for permission to send data using the common transmission medium. A data exchange is carried only between the master device and one selected slave device. The selection is usually done on the basis of an address, which has to be unique for all devices connected to a network.

The RS-485 interface, introduced in 1983, defines differential transmitters and receivers dedicated for symmetric channels. Utilization of symmetric channel eliminates disadvantages encountered in asymmetric transmission channels, wherein due to very low disturbance resistance, the transmission speed and the length of the transmission medium are highly restricted. Three-state transmitters defined by the RS-485 standard are designed for multi-point systems. The typical RS-485 transceiver is shown in Figure 1.

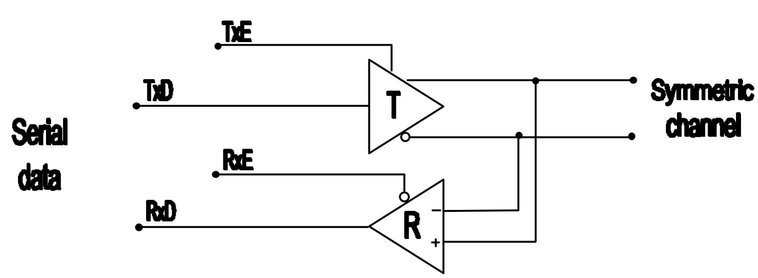


**Figure 1** The RS-485 transceiver

The access to the common transmission medium is controlled by a master device according to the ModBus protocol [1] designed by Modicon. The ModBus protocol is a standard protocol often used in industrial networks and not only in industrial control systems.
The ModBus protocol gives a simple method for medium access but also performs frame error detection, commands confirmation and error reporting. Two groups of devices are controlled by the ModBus protocol. The first group consists of only the master device, which acts as the controller of the common transmission medium. This device decides which device from the second group (slaves) can transmit data. The process of the data exchange that is performed between the master and a dedicated slave device is called transaction. To make possible transaction with chosen slave device, each slave device has to have a unique address. Each transaction comprises of two frames: a query frame sent by the master device and a response frame sent by a slave device. Only the addressed slave device has to response for a query frame. The structure of a query and a response frame is the same and is as shown in Figure 2.

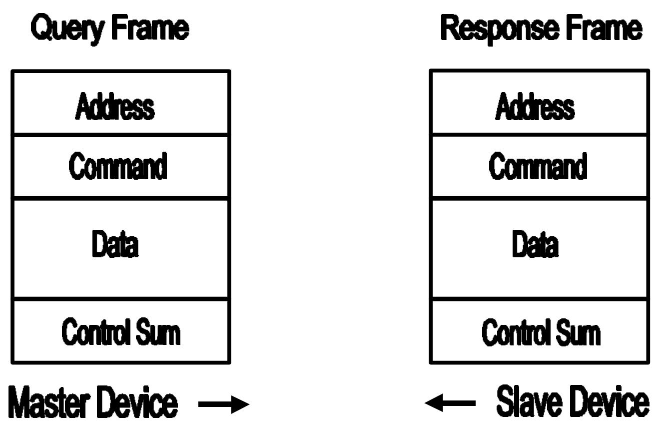


**Figure 2** A query and a response frames in ModBus protocol

ModBus protocol defines two forms of frames, ASCII frames and RTU frames. The difference between them is in the representation of the frame fields. In the ASCII mode, two ASCII characters represent each field of a query and a response frame. The RTU mode uses a binary coding for each element of the ModBus frames. Additional difference is in the control sum and in the way that is used to distinguish the beginning and the end of frames. In the discussed system the ASCII mode of ModBus frames was chosen.

| Begin | Address | Command | Data | LRC Sum | End |
|---|---|---|---|---|---|
| ':' | 2 Chars | 2 Chars | 2xN Chars | 2 Chars | <CR><LF> |

**Figure 3** The ASCII frame

In the ASCII mode two ASCII characters represent each element of a frame. The acceptable characters belong to a set comprising of '0'-'9' and 'A'-'Z'. The address filed consists of the address, which can point to any device from '00' to 'F7' (0-247). The address '00' however has a special meaning. It is used as a broadcast address to pass one request frame to all slave devices at the same time. The command field is filled with a code of the command, which is to be executed by the addressed slave device. In some applications the command field can also confirm proper execution of the command. In this case the most significant bit indicates error if is set to '1', or successful execution of the command when the bit is set to '0'. The next element of the frame is used to send data. In the case the command does not require any data to be sent or received, data field is omitted. The last element of the frame is the control sum, which preserves the contents of the frame. In the ASCII mode the control sum is created as a LRC (Longitudinal Redundancy Check) sum and is appended to the frame as two ASCII characters. The LRC sum is the two's complement of sum of all characters comprising the frame. A receiving device calculates it's own LRC sum of received frame, and then compares it with the received sum. If the sums are the same, frame was received properly, if the sums are different then the frame has to be rejected. When an error occurs, the transaction contains only a query frame since a response frame will not be sent by any slave device due to rejection of the erroneous frame. In this case the master device has to wait only for a certain time called time-out. The time-out parameter sets the maximum time in which a response frame can be sent to the master device. If during this time a frame is not received, the master device can initialize next transaction. Usually the master device repeats the query frame two or more times.

The presented frame shown in Figure 3 is transmitted with beginning and end delimiters. To distinguish the beginning of transmission a frame ':' character (ASCII 3A) is placed before the address field so that, when any device receives ':' character it is expected that next received character will be the first character of the address field. To indicate that frame end, the <CR><LF> (ASCII 0D 0A) characters are put after the LRC sum.

In the given network, the master device, which control data exchange on the bus, is an IBM PC, while the slave devices are measurement devices. The structure of the RS-485 based measurement network is shown in Figure 4.

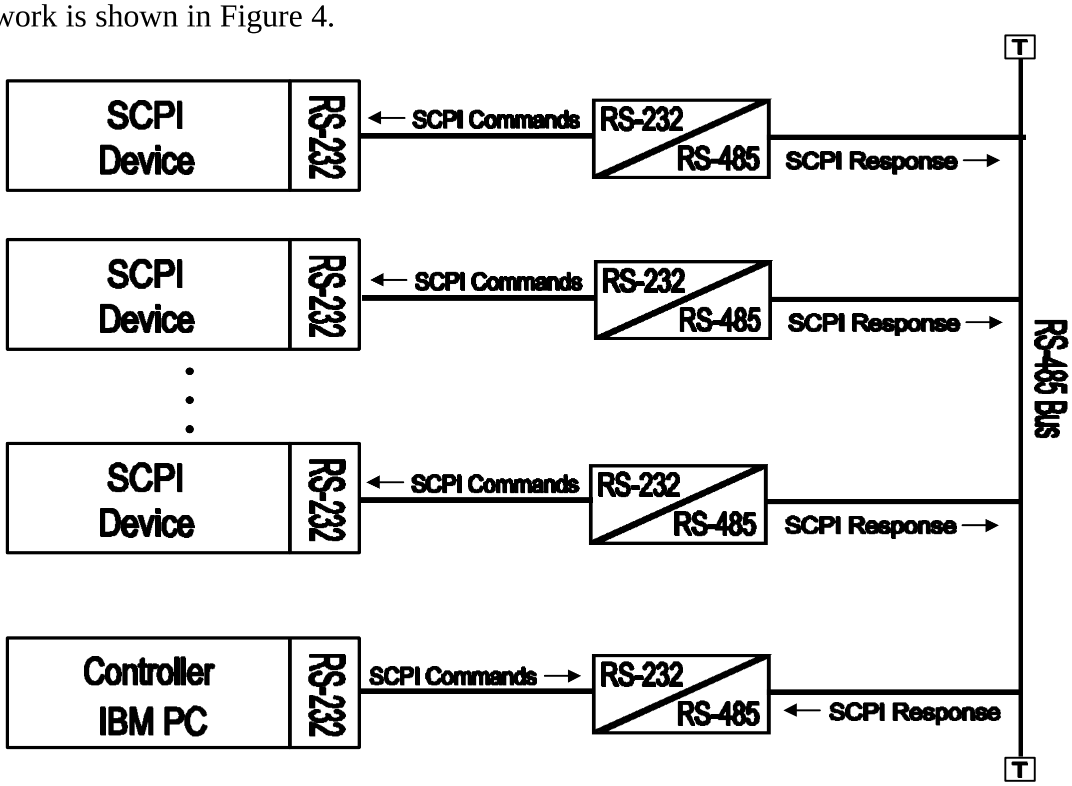


**Figure 4** The RS-485 bus network with SCPI compatible devices

Each device presented in Figure 4 has to cooperate with a special device which converts the signals between the RS-232 and RS-485 interfaces. The RS-232/RS-485 device allows devices with RS-232 interface to have access to the RS-485 bus. The role of the converter is not only translation of the electrical signal between RS-232 and RS-485 interfaces but also control of the access to the RS-485 bus and interpretation of ModBus frames. The RS-232/RS-485 device is a communication device with ModBus protocol implemented in it.

## RS-232/RS-485 COMMUNICATION DEVICE

The RS-232/RS-485 communication device is based on the MCS-51 micro-controller. The communication device [3] is equipped in external I/O bus to which an additional module with RS-232 interface is connected. The RS-485 interface is built using internal serial port of the 80C51 micro-controller. Communication between RS-232 and RS-485 interfaces is controlled by program placed in EPROM memory. The program traces all query frames, which appear on the RS-485 bus looking for the frame with matching address. Each frame is stored into external RAM memory, due to insufficient space in the internal RAM memory of the 80C51 micro-controller. In the shown system maximum frame size was set to 1024 bytes, which is enough for communication with many measurement instruments. To protect devices connected to the common transmission medium, optical isolation is applied at the side of RS-485 interface. The block diagram of the communication device is shown in Figure 5.

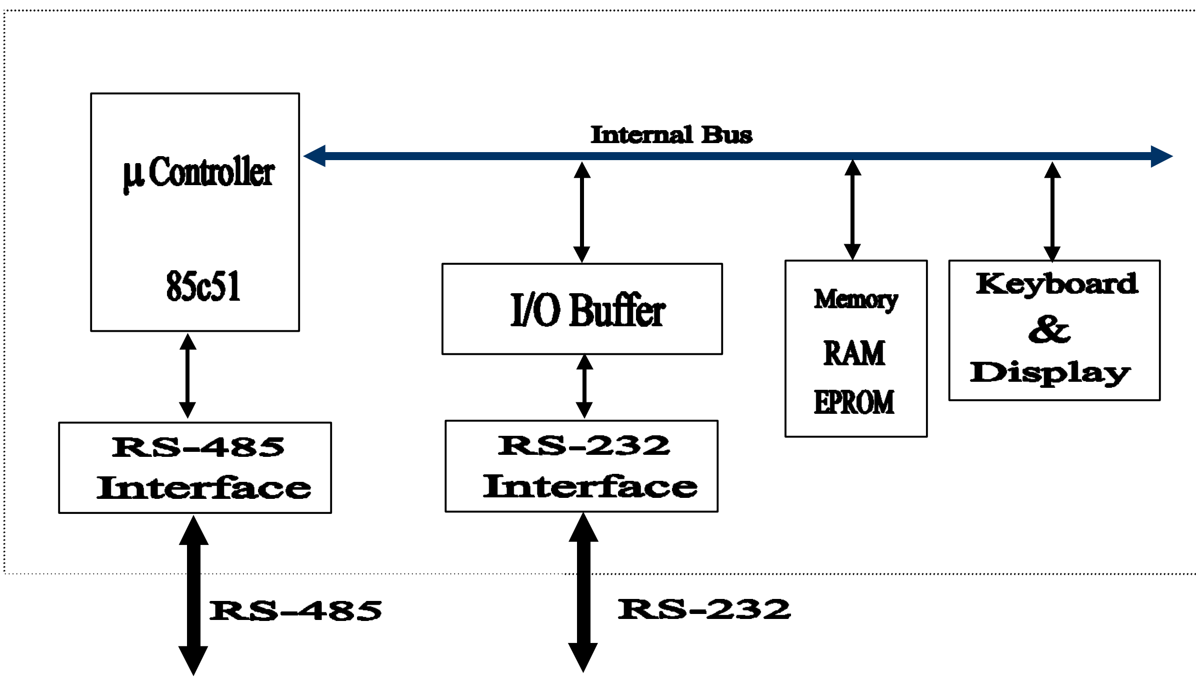


**Figure 5** The RS-232/RS-485 communication device

In Figure 5, the device is connected from one side to the RS-485 bus and from the other side to the RS-232 interface of a measurement device or a controller.

## COMMUNICATION WITH MEASUREMNT INSTRUMENTS

The RS-232 interface is a standard interface of many measurement instruments and is used for remote controlling of the instrument by SCPI commands. The connection between a controller, usually IBM PC, can be realized using three-wire connection or using full modem connection. First option imposes a restriction on data flow control, which can be applied during communication through the RS-232 interface. In this case only XON/XOFF method of data flow control can be applied. To use hardware flow control method additional lines have to be utilized. Usually DTR and DSR lines are used to signal overflow of internal buffers during receiving data.

The SCPI language introduced in 1990 defines programming environment and language used to control measurement devices. The SCPI language describes structure, syntax and commands for communication between a controller and a device in automatic measurement system. The main role of the SCPI language is to determine the form of messages sent by a controller and the form of responses sent back by a measurement device. The SCPI language defines a set of common commands, which have to be implemented in every SPCI compatible devices.

Commands defined by the SCPI standard can be divided into two main groups: commands, which do not require response and commands which do required response from the measurement device. Because the ModBus transaction requires response from selected slave device, the communication device has to prepare response frame regardless of the type of realized command. To ensure proper execution of an SCPI command an additional SCPI command with query about status is sent. The SPCI commands and the responses are ASCII strings ending with <LF> or <CR><LF>.

To simplify and to facilitate uniform communication between elements of the system, the SCPI language format was chosen for sending commands through the RS-485 bus. The ASCII string of an SCPI command is placed into Data filed of a query frame. The RS-232/RS-485 communication device has to pass the command to the measurement instrument through the RS-232 port.

## FUNCTIONS OF THE RS-232/RS-485 COMMUNICATION DEVICE

The RS-232/RS-485 communication device functions as follows:

- receives a ModBus frame from the RS-485 bus,
- interprets the received frame,
- executes received command,
- communicates with a measurement instrument through RS-232 port,
- prepares a response frame, and
- transmits a response frame through RS-485 port.

Each communication device can execute two types of commands, remote commands and local commands. The remote commands are dedicated for the device connected to the local RS-232 port. Usually the remote command passes an ASCII string of an SCPI command to the measurement device. The local commands are dedicated for the communication device and allow for direct communication with the RS-232/RS-485 communication device. Table 1 presents algorithms used in the communication devices.

**Table 1** Algorithms used in the RS-232/RS-485 communication devices

| The communication device connected with a slave device | | The communication device connected with a master device | |
|---|---|---|---|
| 1 | Initialize the RS-232 and RS-485 ports | 1 | Initialize the RS-232 and RS-485 ports |
| 2 | Wait for frame from the RS-485 bus | 2 | Wait for frame from the RS-232 port |
| 3 | Check LRC, if error detected ignore the frame and go to the point 2 | 3 | Prepare the query frame |
| 4 | If received command is a local command execute it, prepare a response frame and go to the point 7 | 4 | Send the query frame |
| 5 | Send the SCPI command(s) through the RS-232 port and wait for response | 5 | Wait for the response frame |
| 6 | Prepare the response frame | 6 | Check LRC, if error detected ignore the frame and go to the point 2 |
| 7 | Send prepared frame through the RS-485 port | 7 | Send content of Data field through the RS-232 port and go to the point 2 |

## HP-VEE ENVIRONMENT

HP-VEE [2] environment is dedicated to create graphical applications for control and measurement systems. The HP-VEE programming is based on the concept of object oriented programming. Each element of the HP-VEE application is an object, which can have inputs, outputs and control terminals. Using terminals, the objects can be connected together thus comprising of a data-flow controlled application.
The HP-VEE environment uses three types of objects to communicate with measurement devices:

- Direct I/O (communication using the hardware interface),
- Component Driver (SCPI commands),
- Instrument Panel (graphical representation of the instrument).

The communication between a HP-VEE application and a measurement instrument is shown in Figure 6.

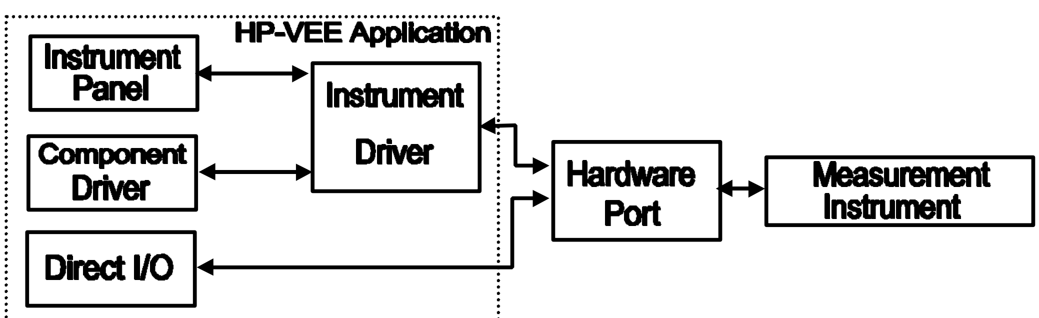


**Figure 6** The communication between HP-VEE application and measurement instrument

The Component Driver and Instrument Panel objects need a special driver dedicated for particular instrument. The driver hides the details associated with SCPI commands and with communication with the instrument. The user does not have to know the SCPI language to use the Instrument Panel object. The Component Driver object gives separate commands realized by the instrument. These two objects cannot be directly used in the described system due to lack of addressing, which is necessary for communication in a multi-point system. The only solution is to adopt the driver [4] by adding one function responsible for addressing. Using Direct I/O functions any ASCII string can be sent through the hardware port. In this case each SCPI command has to be prepared separately and can contain an address needed for the RS-485 bus.

## MEASUREMENT SYSTEM

Measurement system built using described system is shown in Figure 7. The system consists of:

- HP 34401A multimeter,
- HP 33120A generator,
- HP 54600 oscilloscope,
- IBM PC as a controller.

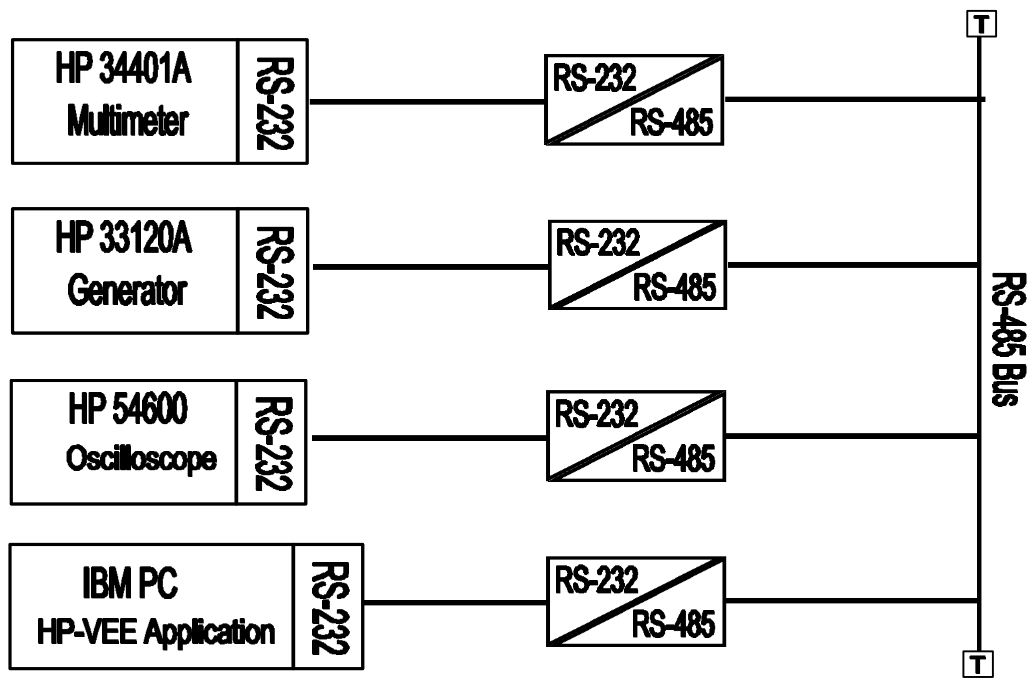


**Figure 7** The Measurement system based on the RS-485 bus

Presented system is controlled by HP-VEE application, which allows for:

- setting the parameters of the signal generated by HP 33120A ,
- setting the parameters of the signal measured by the multimeter,
- controlling of the oscilloscope.

## CONCLUSIONS

Many measurement instruments have a built in serial interface in accordance with the RS-232C standard and the interpreter of the SCPI language. These features provide a possibility of remote control of instruments but only in point to point systems. A design of the measurement system expects a multi-point transmission medium and each of the instruments working in that system must be connected to the transmission medium through an extra device (converter). The converter enables communication between the local RS-232C interface of the measurement instrument and the multi-point RS-485 medium. The bus measurement system, shown on the Figure 4, works under the control of the ModBus protocol that uses a master-slave bus access method. A master device is a personal computer and it controls a bus access, whereas the slave devices (measurement instruments) can only respond on the remote questions of the master device. Designed bus measurement system is shown on the Figure 8 consists of three measurement instruments and a PC, which works as a master device.

## REFERENCES


1. W. Mielczarek: *"Standard Serial Interfaces"*, Helion, Gliwice 1993.

2. R. Helsel: *"Graphical Programming - A Tutorial for HP-VEE"*, Prentice Hall 1995.

3. G. Zar ba: *"The Bus Network of Measurement Instruments Based on the RS-485 Serial Interface"*, 3rd Computer Network Conference, Gliwice 14-15 February 1996. Proceedings published in Informatics (ISSN 0208-7286), No. 30, Gliwice 1996.

4. G. Zar ba: *"The Hierarchical Fieldbus Control in the HP-VEE applications"*, 5th Computer Network Conference, Gliwice 18-19 March 1998. Proceedings published in Informatics (ISSN 0208-7286), No. 34, Gliwice 1998.