Annals of the University of Petrosani, Electrical Engineering, 23 (2021)

USING SENSORS WITH MODBUS COMMUNICATION RTU INTO A DATA ACQUISITION SYSTEM

NICOLAE PĂTRĂȘCOIU[1], COSMIN RUS[2], CECILIA ROȘULESCU[3] , NICOLETA NEGRU[4]

Abstract: In this paper we propose an application written in LabVIEW through which a number of transducers or sensors are controlled so that the data from them are taken according to a certain algorithm and also this application realize the data processing. Control consists in selection for each of these transducers, their inputs selection and retrieves information from them. The physical connection between used transducers and master is made through RS-485 bus and the transmission protocol used is Modbus RTU.

Keywords: RS-485 interface, environmental sensors, Modbus RTU, LabVIEW.

1. INTRODUCTION

It is known that a data acquisition system can be defined as a measurement system that allows both visualization and recording of the evolution over time of several analog or numerical quantities.

The traditional operator's control function has been replaced by instruments and sensors that give very accurate measurements and indications, making the control function totally operator-independent. The processes can be fully automated. Instrumentation and sensors are an integral part of process control, and the quality of process control is only as good as its measurement system [1].

The instrumentation and sensors send the collected data, representing the controlled quantities' values through electrical signals. The current practice involves their digital transmission, that's means the transfer of binary codes to the equipment that allows the visualization or processing of this acquired data.

Here are many solutions both in terms of the hardware configuration of the data transmission structure and software solutions for their implementation. One of

[1] *Associate Prof. Eng. Ph.D. at the University of Petroşani, nicolaepatrascoiu@upet.ro*
[2] *Eng.,Ph.D. student at the University of Petroşani, cosminrus@upet*
[3] *Professor, Grigore Geamănu School, ceciliarosulescu@yahoo.com*
[4] *Ph.D. student, Systems engineering, University of Petrosani, nicoletanegru@upet.ro*

these possibilities is to use the RS-485 serial transmission bus together with the Communication Modus protocol [2].

Data collection, from different environments, can be done by manual measurements and their subsequent manual or computerized processing [3]. Their real-time data acquisition brings a new perspective to the scientific analysis of the information contained [4].

Different methods and structures are been developed following the different aspects of environmental monitoring systems. Thus a very important aspect is related to the distance between the measurement points and the data processing location. If these distances are large, different radio technologies are used for data transmissions, such as GSM and wireless standards as IEEE 802.15.4 or ZigBee protocols [5].

However, if these distances are of the order of tens or hundreds of meters, it is possible to use transmission technologies using various physical environments. This second variant has the advantage that it also offers the possibility of remote powering of nodes and sensors.

The solution that we propose through this paper comes to complete the known solutions, by the fact that the data collection is done according to a specific algorithm. This algorithm aims to obtain functional characteristics of the input-output type based on predefined values of the input quantities. Thus, automatic measurements are performed and graphs of the evolution of the quantities subject to observation under predetermined conditions are provided to the user.

The algorithm is validated by means of a laboratory stand in which various transducers are used for various physical quantities connectable on the RS-485 bus with data transfer via Modbus RTU protocol.

2. HARDWARE STRUCTURE

The used sensors are a family of environment sensors with an RS-485 interface and these communicate using standard Modbus RTU protocol. We used the Tibbo four types of sensors:

1. temperature sensor (BP#01),
2. combined temperature and humidity sensor (BP#02),
3. light sensor (BP#03),
4. 3-axis accelerometer (BP#04).

All of these are typically wired to a twisted pair cable, which distributes power and carries RS485 "+" and "-" lines and sensors accept the supply voltage in the 4V to 15V range.

Each of the 4 sensors used in the acquisition system is made using the measurement integrated circuits specific to the controlled quantities. The structure circuits of the four sensors used, as well as their main measurement parameters, are presented in the table 1 [6]:

Table 1. ***The structure circuits of the four sensors used***

Type of sensor	Measurement IC	Measurement specifications
BP#01 – ambient temperature sensor	MCP9808 Microchip	➢ Measurement range: -40°C to + 125°C ➢ Measurement resolution: 0.25°C ➢ Measurement accuracy: ± 0.5 °C
BP#02 – ambient temperature and humidity sensor	HIH6130 Honeywell	1. <u>Temperature measurement</u> ➢ Measurement range: -25°C to 50°C ➢ Measurement resolution: 0.5°C ➢ Measurement accuracy: ± 0.5 °C 2. <u>Humidity measurement:</u> ➢ Measurement range: 10 to 90% RH ➢ Measurement resolution: ±0.1% RH ➢ Measurement accuracy: ±5% RH4 ➢ Temp. range for valid humidity measurements: 5°C to 50°C
BP#03 – ambient light sensor	BH1721FVC Optical Sensors	➢ Measurement range: 1 to 65528 lux ➢ Measurement resolution: 1 lux ➢ Measurement accuracy: 1 lux
BP#04 - 3-axis accelerometer	ADXL312 Analog Devices	➢ Independent X, Y, and Z axes ➢ Measurement range for each axis: ±6G ➢ Measurement resolution for each axis: 0.003G ➢ Measurement accuracy for each axis: 0.1G

Figure 1 shows the sensor capsule, with the specification that it is common to all 4 types of sensors used.

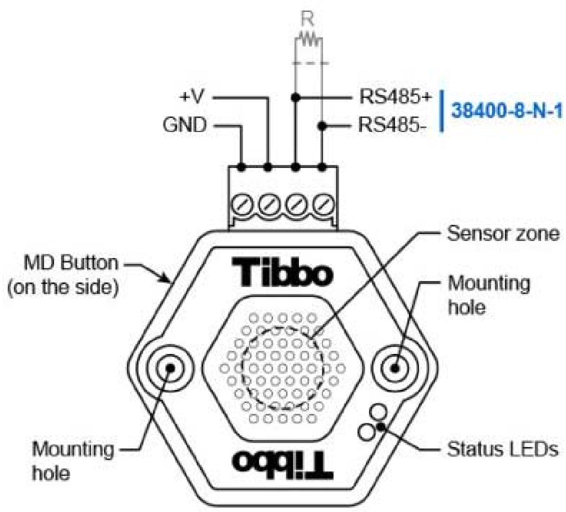

Fig.1. The connector and control elements of the used sensor

The RS-485 applications use data communication over twisted-pair cable because the noise from external sources couple equally into both signal lines as common-mode noise, which is rejected by the differential receiver input.

Using only one signal pair means half-duplex communication mode and that implies the data driving and receiving to occur at different times and necessitate the controlled operation of all nodes via direction control signals, such as Transmitter/Receiver Enable signals.

RS-485 standard conforms transmitters provide a differential output of a minimum 1.5 V across a 54-Ω load, whereas standard conform receivers detect a differential input down to 200 mV.

The two values provide a sufficient margin for a reliable data transmission even under severe signal degradation across the cable and connectors. This robustness is the main reason why RS-485 is well suited for long-distance networking in a noisy environment.

Based on these considerations is build the experiment configuration with the sensors connected to the RS-485 bus and this configuration is shown in Fig. 2

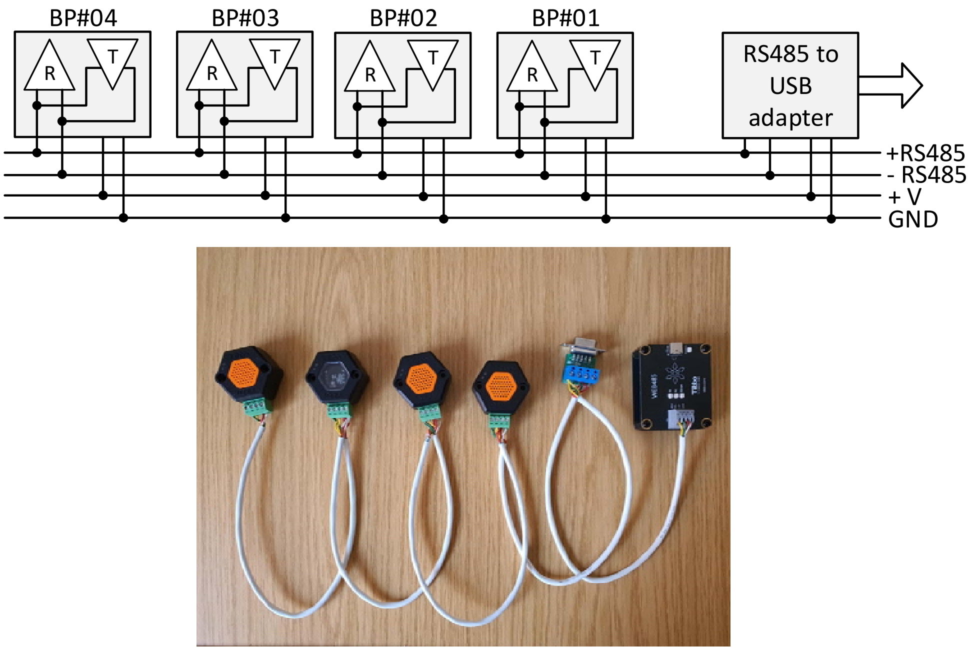

Fig.2. The diagram and the setup of the experiment.

Because the sensors, which represent the slave devices, communicate with a master device or equipment through RS-485 serial bus and when the master device is a PC or laptop they should have a compatible port for connection to this serial bus.

The current and usual PC or laptop communication port is the USB port and in these cases for connecting the sensors is required a USB to RS-485 adapter. In developing this application is used the UT-850 converter (a Data Communications product), which ensures a data transfer rate of up to 921.6 Kbps. This RS-485/422 to USB converter adds a virtual serial port on a desktop or laptop so that the outputs are automatically configured as additional COM ports [7], [8].

Parameters of the data communication through RS-485 bus are:

- Half-duplex (two-wire) RS-485 interface
- Communications parameters: 38400-8-N-1
- Modbus RTU protocol
- Sensors powered via the USB port and RS-485 adapter [9]

3. SOFTWARE IMPLEMENTATION

The software used to control the data acquisition from the sensors and also the data processing that we propose in this paper is built in the LabVIEW graphical programming environment and like all applications builds in LabVIEW is named virtual instrument.

A virtual instrument has three components, i.e., a front panel that is the user interface, a block diagram that is the proper program and icon used in the hierarchical structure of the virtual instrument.

The virtual instrument build for this application consists of the main program and a number of subprograms, called SubVI used to simplify the diagram block configuration.

This programming environment was chosen due to its ease of use, the multitude of functions and libraries available and even due to the fact that in the current context of the development of IoT technologies there are more and more implementations of specific IoT programs in LabVIEW. Also, this programming environment is widely used in terms of creating hardware and software platforms that allow the development of online laboratories both in the IoT context and especially in the current pandemic context.

3.1. Front panel

The front panel consists of three components through which the user can interact with the virtual instrument.

The first component is dedicated to the necessary settings for communication with the sensors on the bus via the Modbus protocol. The parameters are set according to the previous specifications [9].

The second component is used for the selection of one of the two working modes, namely: manual query (acquisition), figure 4, or automatic query (acquisition), figure 5. For the manual query, the user has a TAB control to choose the sensor to be read (BP # 01 ... BP # 04).

For the automatic query, all the 4 sensors will be interrogated cyclically, thus 7 values of the measured quantities will be displayed simultaneously: temperature from the BP # 01 sensor, temperature and humidity from the BP # 02 sensor, light intensity from the BP # 03 sensor, as well as the movements along the X, Y and Z axes from the BP # 04 sensor (Giro).

The third component offers the user the processed data in the form of a graph with every quantity evolution in time for the manual query, (figure 3) respectively these quantities in the form of a table which is displayed all the values and the time stamp of their acquisition (figure 4).

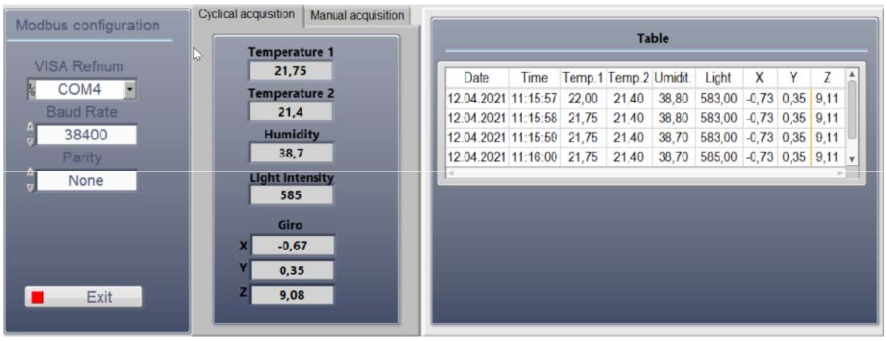

Fig.3. Front panel in the Manual Acquisition mode

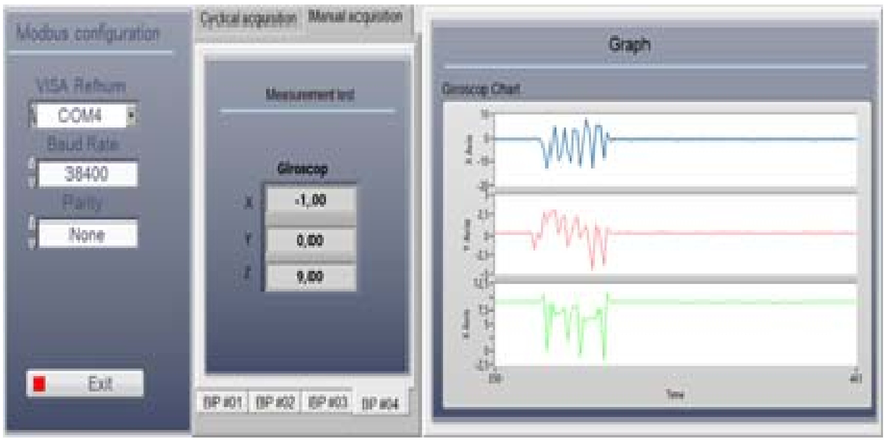

Fig.4. Front panel in the Cyclical Acquisition mode

3.2. Block diagram

The block diagram represents the program itself and is built with operations, functions, and programming structures from LabVIEW libraries.

The main programming structure in making the virtual instrument is a **WHILE** loop that makes the program work until the user presses the **STOP** button. While running the program, the user can always switch between the two operating modes of the virtual instrument, namely, **Manual Acquisition** mode or **Cyclic Acquisition** mode.

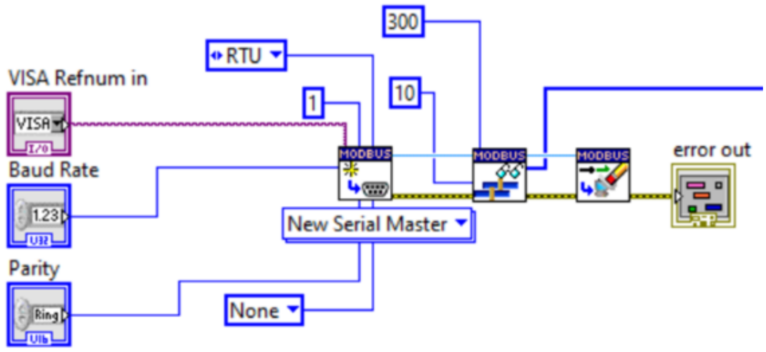

Fig.5. Modbus reading functions

Switching between the two working modes is performed by means of a Case structure controlled with **TAB Control**.

For the **Manual Acquisition** mode, the user has another tab control, **Select Device**, with which he can select any of the four sensors on the bus. The selection is made by the ID of each sensor and to read the information available to it is chosen is used the list of Modbus registers for each sensor. Based on the two identifiers, subVIs are made for each of the 4 sensors using Modbus functions from the NI Modbus library, as shown in fig.5.

The measured values are thus taken over and transmitted to **Chart** type indicators to represent their evolution over time. Each graphical indicator is activated with the selection of the respective sensor and for this their **Property** type nodes are used with the choice of the **Visible** property. The block diagram of the virtual instrument for the operation corresponding to the **Manual Acquisition** mode is presented in figure 6.

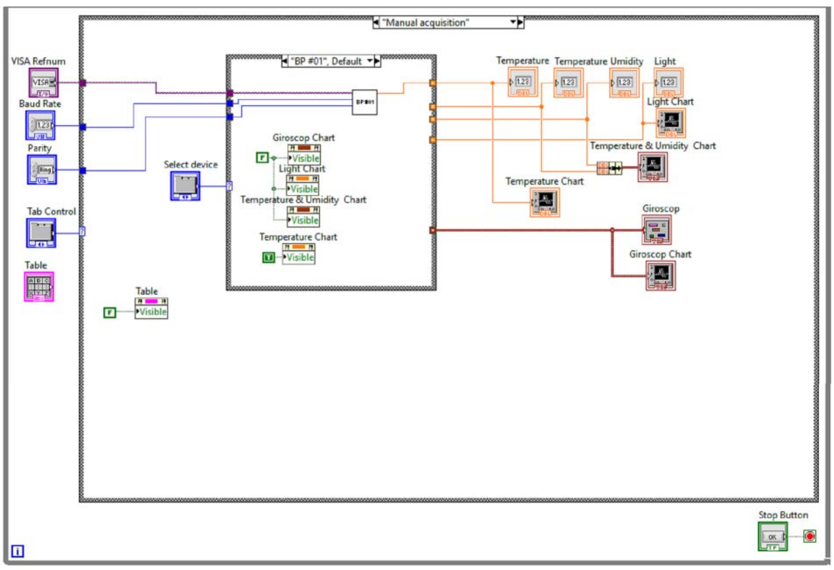

Fig.6. Block diagram of the **Manual acquisition** mode

By choosing the **Cyclic Acquisition** operation mode on the front panel, indicators for each size will be available and their values are read by including a **FOR** loop in the structure of the block diagram. The number of cycles required to read the sensors is set so that constant 4 is connected to the **Count terminal**. The block diagram of the virtual instrument for the operation corresponding to the **Cyclical Acquisition** mode is presented in figure 7.

The selection of each sensor is made by using the iteration variable of this structure which also controls a **CASE** structure by connecting that to the **Case selector**. Thus, in each cycle, the iteration variable is incremented and, by default, the next sensor is selected. This selection procedure is resumed after each reading package of all sensors.

The iterative reading of each sensor is done by calling the same SubVI used in the manual acquisition mode.

Because the values read on the bus correspond to a single sensor at a time, all other sensors send the value 0 at that time, in order to retain all the values, read in a cycle of the **FOR** loop, a displacement register associated with this **FOR** structure is used.

The values read from each sensor are taken through local variables and converted from numeric format to string format via the **Number to Fractional String Function**. These functions convert a number to an F-format (fractional notation), floating-point string at least width characters wide or wider if necessary.

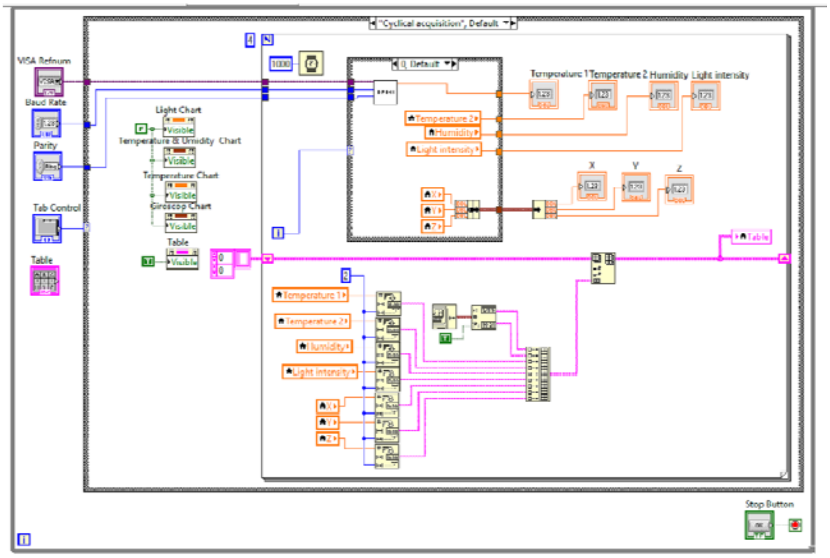

Fig.7. Block diagram of the **Cyclical acquisition** mode

The strings representing the read values are added with the information about the time (date and time) of their acquisition obtained with the **Gate Date/Time in Seconds** function and all are grouped in an array using the **Build Array** function.

Each new data packet is inserted into the array containing data from previous acquisitions, retrieved through the **Shift Register**, using the **Insert into Array** Function. All these data are sequentially displayed in **Table**.

Although it has imposed constraints since its inception more than 40 years ago, related to the speed of data transmission, the maximum number of devices that can be interconnected and possible problems caused by potential sources of interference, the Modbus protocol is widely used, especially in energy consumption monitoring applications and as seen in the present case and in terms of managing a network of sensors].

Even though there is more and more talk these days about wireless sensor networks (WSN), wired sensor networks have not yet reached their maximum potential and are still widely used, especially in environments where it is not possible to provide good protection against any type of interference. Thus, we will always try to find the simplest and reliable solutions to improve technologies that allow the acquisition of various data from different sensors to achieve high-performance monitoring systems, especially in the context of new IoT technologies.

4. CONCLUSIONS

This paper presents its own contribution related to the Modbus communication protocol and its use in a network of sensors. Thus, through its own algorithm implemented in LabVIEW, a manual and automatic reading of 4 types of sensors was achieved (temperature sensor, temperature and humidity sensor, light sensor and accelerometer). In the manual operation mode of the program, it allows the user to select the type of sensor from which he wants to take the data so that later they can be processed and displayed as a graph. In the automatic reading mode, an automatic reading of all the sensors is performed and the results are displayed in the form of graphs specific to each sensor.

Thus, an overview can be made of some important physical parameters in a certain area proposed to implement such a monitoring system using a network of sensors that use the Modbus communication protocol.

The hardware structure can be expanded to 255 sensors with the appropriate settings for their identifiers. The length of the bus, according to RS485 specifications, can be extended up to 1200 meters, which ensures a relatively large coverage area. Any addition of sensors in the structure implies a minimal intervention in the program regarding only the number of sensors that will be read either manually or cyclically.

The entire system of acquisition, processing and display of results implemented and presented in this paper has a special reliability being tested both in the laboratories of the University of Petrosani and in external environmental conditions and has proven its effectiveness and accuracy.

The major advantage of this system is that it can be included extremely smoothly and quickly in other specialized programs due to its modular structure, the entire program is built around a central core that deals only with data processing while for communication with the four sensors four subprograms (SubVI) are used which can be modified or used as desired.

REFERENCES

[1]. Căldăraru F., Căldăraru M., *Metode de măsurare şi monitorizare a parametrilor de calitate a mediului*, Editura Cavallioti, Bucureşti, 2010.

[2]. Derenzo E.S., *Practical Interfacing in the Laboratory Using a PC for Instrumentation*, Data Analysis, and Control, Cambridge University Press, New York, 2003.

[3]. Draghici C., Perniu D., *Poluarea si monitorizarea mediului*, Ed. Univ. Transilvania, Brasov, 2012.

[4]. Jafer E., Spinar R., Stack P., O'Mathuna C., Pesch D., *Wireless sensor network deployment for building environmental monitoring and control*, AMBI-SYS, Berlin, 2011.

[5]. Nikhade S.G., , Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications, ICSTM, India, 2015.

[6]. Rus C., Negru N., Patrascoiu P., *Low-cost System to Acquire Environmental Parameters in Urban Areas in the Context of IoT*. Journal of Environmental Protection and Ecology 20, No 3, 1451–1461, 2019.

[7]. *** https://tibbo.com/store/rs485-modbus-sensors.html

[8]. *** *Modbus Application Protocol Specification*. http://www.Modbus-IDA.org, 2006.

[9]. *** *RS485 Modbus Sensors ("Bus Probes")*. https://docs.tibbo.com/phm/bp