

# Industrial area environmental monitoring based on transducers with MODBUS communication

Nicolae Patrascoiu<sup>1\*</sup>, Cosmin Rus<sup>1</sup>

<sup>1</sup>University of Petrosani, Department of System control, computer science electrical and power engineering, 20 Universitatii, Petrosani, Romania

**Abstract.** The monitoring of environmental parameters in industrial areas where potential sources of pollution exist is very important from the point of view of prevention of environmental accidents. In this paper, we propose a solution for the monitoring of the environmental parameters with the local acquisition through specific environmental and movement sensors and data transmission to a higher hierarchical level through the use of MODBUS communications. A flexible hardware structure and software development concept are presented to offer local information and to be integrated into an environmental quality monitoring network.

## 1 The issue of environmental monitoring

Environmental monitoring is a set of complex activities aimed at obtaining data on the environment's quality and their processing. Obtaining data involves systematic measurements and long-term and systematic environmental parameters with spatial and temporal coverage, organized to ensure decision-making concerning the objectives pursued. In general, environmental monitoring involves tracking changes in environmental "properties" recorded as a consequence of pollution and/or other factors that influence natural or anthropogenic factors. Thus, different types of monitoring have been delimited and further developed, such as [1]:

- geophysical monitoring - supervision of geophysical changes;
- chemical monitoring - monitoring changes in chemical composition that occur in the environment;
- physical monitoring - surveillance of physical properties (temperature, humidity, moisture, lightning, mechanical changes, noise, radioactivity);
- biochemical monitoring - surveillance of biochemical changes.

The information obtained by measurement provides the basis for decision making and development of environmental management strategies and the existence of an environmental monitoring system results from the following needs [2]:

- To know the evolution of the quality of environmental factors to establish and impose measures for protection, conservation, refurbishment, and verification of the effectiveness of measures taken.
- To group, select and order information and correlate it with other information.

---

\* Corresponding author: [nicolaepatrascoiu@upet.ro](mailto:nicolaepatrascoiu@upet.ro)

- To obtain information comparable to information on a regional or global scale for their use in their own programs or international programs (global climate, ozone layer, etc.).
- To know and quickly assess the situation in cases of accidents or anthropogenic incidents.
- To develop the knowledge base for establishing actions for environmental protection and ecological reconstruction.

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.

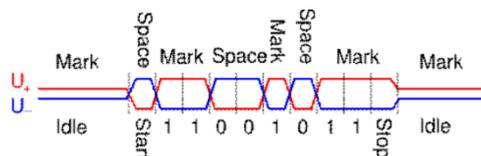
## 2 Hardware structure

The configuration of the monitoring system of the environmental parameters that he proposes in this paper involves the use of sensors produced by the company Tibbo Technology which is focused on designing, manufacturing, and marketing programmable IoT modules, boards, and controllers for the Internet of Things (IoT), Industrial Internet of Things (IIoT), and automation applications [6].

The used sensors represent a family of environment sensors with an RS485 interface that communicates the desired data using the standard Modbus RTU protocol. The used sensors to build the monitoring system test are the ambient temperature sensor (BP#01), ambient temperature and humidity sensor (BP#02), ambient light sensor (BP#03), and 3-axis accelerometer (BP#04).

### 2.1 RS-485 bus

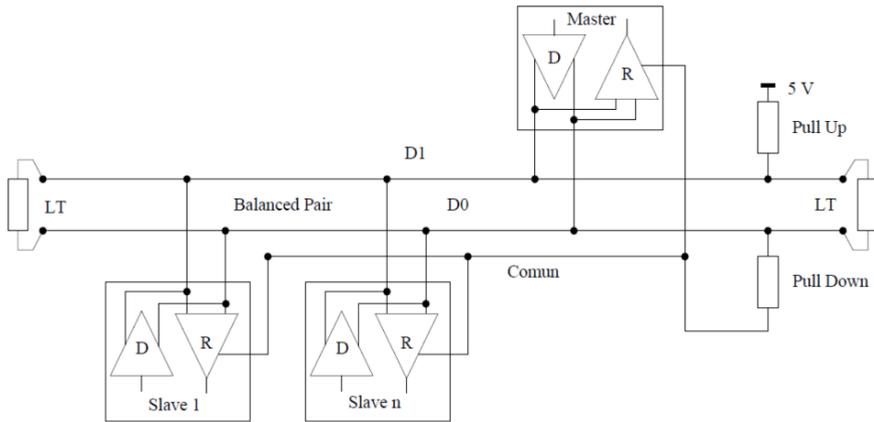
The RS-485 bus is a relatively long-distance serial communication system allowing communications up to 1200 m. Unlike other standards that transmit the value of a bit through the voltage level in the case of the EIA / TIA-485 standard, the bit differential coding mode is used, so the signal appears between two wires that transmit only the signal, they being different from the electric mass ( Fig. 1.).



**Fig.1.** Signal waveform on the RS-485 bus

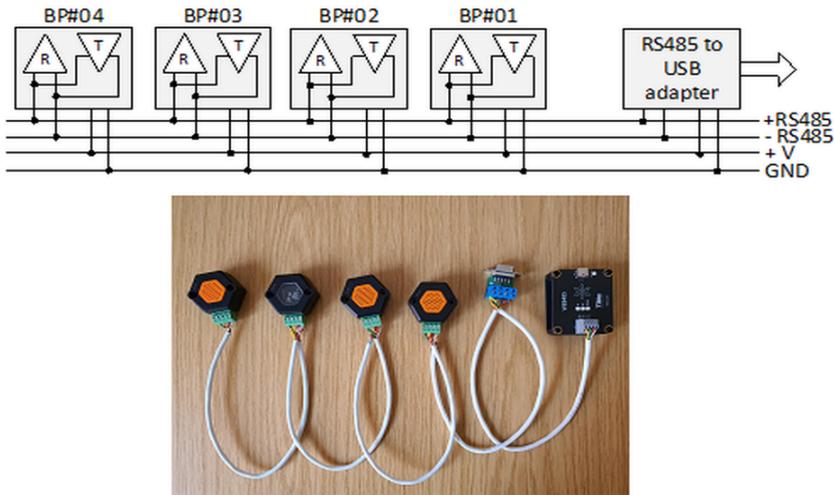
Network topologies can be point-to-point, point-to-multipoint, and multipoint - multipoint. The maximum number of transmitter-receiver pairs can be 32. The common mode voltages they support are extended from +12 to -7 V due to the capabilities of using the high impedance state (tri-state).

The general structure of a semi-duplex configuration of two-wire serial data transmission via the RS-485 bus is shown in fig.2. As in the case of this communication bus, the use of differential transmission lines imposes the need to close their circuit by using terminal elements, LT. These are usually electrical resistors and are realized in the sensor's circuit.



**Fig.2.** The general structure of the RS-485 communication bus

The connection of the sensors, which are a slave on the bus, used into a point-to-multipoint network topology on the RS-485 bus is shown in fig.3. The role of the master is assigned in this application to the computer.



**Fig.3.** The sensors connection diagram and the setup of the laboratory experiment

The sensors communicate with a master device or other 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. Now the usually communication port is the USB port and it is highly unlikely that a PC and a laptop have an RS485 port [7].

In these cases for connecting a RTU device is required an USB to RS485 adapter. In developing this application we used the UT-850 converter (a Data Communications product). This is a converter RS-485/422 to USB that adds a virtual serial port on desktop or laptop. Serial outputs are automatically configured as additional COM ports.

## 2.2 Modbus protocol

The Modbus is one of the most popular industrial protocols being used today, for good reasons like simple, inexpensive, universal, and easy to use.

This protocol was developed by Modicon (now Schneider Electric) in 1979 as a means for communicating with many devices over a single twisted pair wire. If the original was built to run over the RS-232 bus, it was adapted to run on the RS-485 to gain faster speed, longer distances, and a true point-to-multipoint network.

Modbus is a master-slave communication system, where the master communicates with one or multiple slaves devices. The master can be a Programmable Logic Controller - PLC, a PC, a Distributed Control System - DCS, or a Remote Terminal Unit - RTU. The slave devices on Modbus networks cannot initiate communication; they can only respond.

The three Modbus versions are used Modbus ASCII, Modbus RTU, and Modbus/TCP. All Modbus messages are sent in the same format and the only difference among the three Modbus types is in how the messages are coded [8].

In this application, we use the Modbus RTU version where data is coded in binary and requires only one communication byte per data byte at speed 38400 baud. This is ideal for use over point to multipoint RS485 bus networks, at speeds from 1,200 baud to 115 Kbaud.

To make a connection to reading a slave device the master sends to that a message which contains: device address, function code, reading data command, and the error check.

Parameters of the data communication through RS-485 bus are [6]: Half-duplex (two-wire) RS-485 interface, Communication parameters: 38400-8-N-1, Modbus RTU protocol, Sensors powered via the USB port and RS-485 adapter

The Modbus Registers of the used sensors presented in **Table 1** and the structure circuits of the sensors used, are presented in the **Table 2**.

**Table 1.** The Modbus Registers

Addr.	Name	R/W	Type	Description
200	ID (non-volatile)	R/WP	Unsigned 16-bit	Device ID (Modbus address). This ID must be between 1 to 247. Attempts to set an ID outside of this range will be ignored. Default ID of a Probe depends on the Probe's type: 0x40 for BP#01 (ambient temperature sensor); 0x41 for BP#02 (relative humidity and temperature sensor); 0x42 for BP#03 (ambient light sensor); 0x43 for BP#04 (3-axis accelerometer).
201	TYPE	R	Unsigned 16-bit	Probe type: 1 — BP#01 (ambient temperature sensor); 2 — BP#02 (ambient temperature and humidity sensor); 3 — BP#03 (ambient light sensor); 4 — BP#04 (3-axis acceleration sensor).
203	SIGNATURE	R	Unsigned 16-bit	Always returns 0xAA55. This is a common signature for all Bus Probes.
207	USER WORD 1 (non-volatile)	R/W	Unsigned 16-bit	Users are free to save any application-specific data in this register.
208	USER WORD 2 (non-volatile)	R/W	Unsigned 16-bit	Users are free to save any application-specific data in this register.
209	USER WORD 3 (non-volatile)	R/W	Unsigned 16-bit	Users are free to save any application-specific data in this register.
210	USER WORD 4 (non-volatile)	R/W	Unsigned 16-bit	Users are free to save any application-specific data in this register.

**Table 2.** The parameters of the sensors used

Type of sensor	Modbus 16 bit data register address	Measurement IC	Measurement specifications
BP#01 – ambient temperature sensor	300	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	400 404	HH6130 Honeywell	<b>1. <u>Temperature measurement</u></b> Measurement range: -25°C to 50°C Measurement resolution: 0.5°C Measurement accuracy: ± 0.5 °C <b>2. <u>Humidity measurement:</u></b> 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	500	BH1721FVC Optical Sensors	Measurement range: 1 to 65528 lux Measurement resolution: 1 lux Measurement accuracy: 1 lux
BP#04 - 3-axis accelerometer	600 602 604	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

### 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 built 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

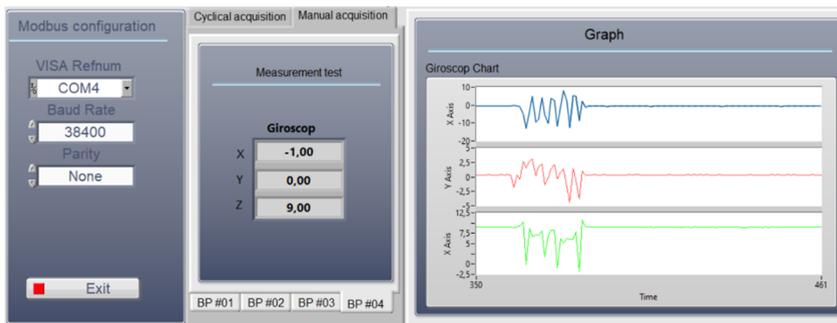
The front panel consists of three components through which the user can interact with the virtual instrument.

The first component is dedicated to the necessary settings for communication with the

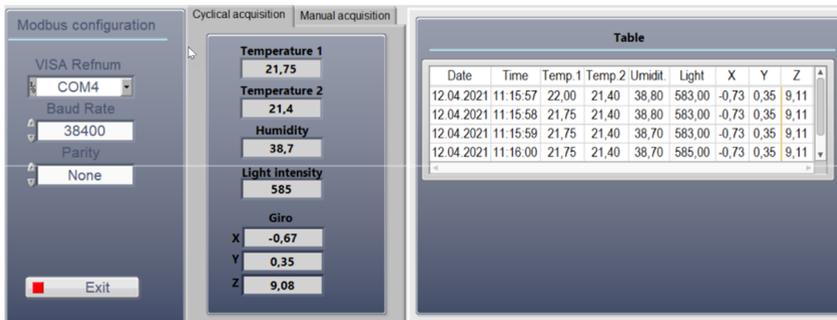
sensors on the bus via the Modbus protocol. The parameters are set according to the previous specifications [9].

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 4) respectively these quantities in the form of a table which is displayed all the values and the time stamp of their acquisition (figure 5).



**Fig.4.** Front panel in the **Manual Acquisition** mode



**Fig.5.** Front panel in the **Cyclical Acquisition** mode

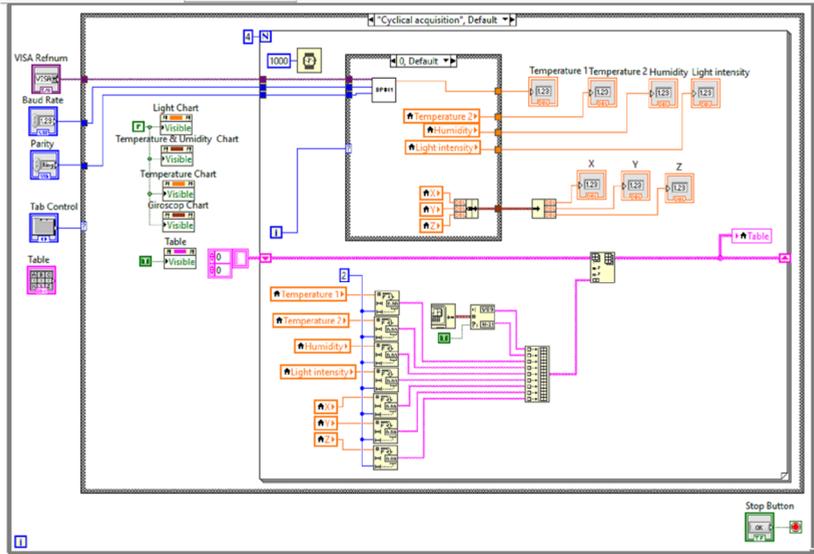
### 3.2. Block diagram

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.

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





**Fig.8.** Block diagram of the **Cyclical acquisition** mode

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.

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.

## 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. F. Căldăraru, M. Căldăraru, *Metode de măsurare și monitorizare a parametrilor de calitate a mediului* (Editura Cavallioti, București, 2010)
2. C. Draghici, D. Perniu, (2002) *Poluarea și monitorizarea mediului*. (Ed. Univ. Transilvania, Brasov, 2012)
3. C. Rus, N. Negru, P. Patrascioiu, *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)
4. E. Jafer, R. Spinar, P. Stack, C. O'Mathuna, D. Pesch, AMBI-SYS 2011, *Wireless sensor network deployment for building environmental monitoring and control*, (Berlin, 2011)
5. S.G. Nikhade, ICSTM 2015, *Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications*, (India, 2015)
6. <https://tibbo.com/store/rs485-modbus-sensors.html>
7. E.S. Derenzo, *Practical Interfacing in the Laboratory Using a PC for Instrumentation, Data Analysis, and Control*, (Cambridge University Press, New York, 2003)
8. Modbus Application Protocol Specification. <http://www.Modbus-IDA.org> (2006)
9. RS485 Modbus Sensors ("Bus Probes"). Available on <https://docs.tibbo.com/pbm/bp>