# DATA MONITORING OF SENSORS IN A BUILDING USING THE LABVIEW GRAPHICAL PROGRAMMING ENVIRONMENT

Sebastian-Alexandru Dragusin[1], Rodica-Mihaela Teodorescu[2], Grigore-Adrian Iordachescu[3]
Department of Computer Science
University of Pitesti
Pitesti, Romania
[1]dragusin.sebi@yahoo.com, [2]mihrodteo@gmail.com, [3]adi_iord@yahoo.com

Keywords- *sensors, environmental parameters, virtual instrument, Arduino microcontroller, LabVIEW*

Abstract: *This paper presents the design of an application for viewing the environmental parameters (temperature, humidity, carbon monoxide level) of an enclosure, using the Arduino microcontroller and the LabVIEW graphic design environment. Sensors have been programmed with the Arduino microcontroller and their return values can be actually visualized through the virtual tool. The virtual instrument was implemented using the graphical programming environment Labview, its front panel looking like a real measuring instument. The controls and indicators on the front panel have the appearance and functionality similar to real ones and can be easily set and selected with the mouse. This virtual instrument could be easily improved both in appearance and functionality, depending on the needs of its user.*

## I. INTRODUCTION

LabVIEW is a program widely used for monitoring different sensors, given the number of works published in this regard. An example would be a scientific paper which presents in real time a virtual tool for visualizing the characteristics of the photovoltaic panel [1]. Alternatives to LabVIEW for monitoring data from ambiental sensors are Raspberry Pi boards or ESP microcontrollers used together with a web graphical interface.

Before the advent of modern technology, the human reading of surrounding external information was initially based on the five sense organs. Sensors, by the very Latin origin of the word, lead one to the meaning of perceiving, feeling.

The initial sensor synthesis was realized on this natural basis and a first sensor classification can be made according to this reasoning:

- seeing → optical sensors;
- smelling → gas and humidity sensors;
- hearing → acoustic sensors;
- touching → pressure sensors [2].

A tranducer, in general, is constructed using two components: the sensitive element and the adapter.

The sensor, or the sensitive element, represents the specific element for detecting the physical quantity that the transducer has to measure (let that quantity be called “x”).

The state of the sensitive element is modified under the action of the input quantity „x” as a consequence of some physical laws. This state will thus contain the necessary information to determine the value of the initial „x” quantity [3].

The sensor characteristics can be divided in three cathegories:

- Input characteristics
- Transfer characteristics
- Output characteristics

The following input characteristics of sensor can be found in the technical literature [4]:

- Range: represents the minimum and maximum value of the sensed and measured physical quantity. For the example of a Resistance Temperature Detector (RTD) temperature can be measured in the interval of -200 to 800 °C.
- Span: represents the amplitude of the range as the difference between the maximum and the minimum values of the measured quantity. In the above example, the span of RTD is 800- (-200) = 1000 C.
- Accuracy: represents the difference between the measured value and the real (true) value.

- Precision: represents the closeness among a set of measured values.
- Sensitivity: represents the ratio of change at the output compared to the change at the input.
- Linearity: represents the maximum deviation between the measured values of a sensor from an ideal liniar curve.
- Hysteresis: is the difference in output when input is varied in both increasing and decreasing directions.
- Resolution: is the minimum change in input that can be sensed by the sensor.
- Reproducibility: is defined as the ability of a sensor to produce the same output when an identical input is applied.
- Repeatability: is defined as the ability of a sensor to produce the same output when an identical input is applied and all the physical and measuring conditions are kept the same (including the operator, instruments, environmental conditions).
- Response Time: is generally expressed as the time at which the output reaches a certain percentage (for instance, 95%) of its final value, in response to a step change of the input quantity.

The sensors are classified according to several criteria [5]:

- according to the nature of the input quantity to be measured, the following types of sensors can be distinguished:
  - movement sensors;
  - temperature sensors;
  - radiation sensors.
- in function of the output quantity:
  - resistive sensors and transducers;
  - capacitive sensors and transducers;
  - inductive sensors and transducers.
- in function of the mode for obtaining the energy required to form the metrological signal:
  - active sensors;
  - pasive sensors.
- in function of how the output quantity varies:
  - analog sensors;
  - digital sensors .
- by the shape of the static characteristic:
  - liniar transducers
  - non-liniar transducers.

## II. HARDWARE

The sensors used to construct the device discussed in this paper are presented below:

- Temperature sensor LM35. The temperature sensor is a device for measuring the temperature by an electrical signal generated by a thermocouple or RTD (resistance temperature detector);
- Soil moisture sensor. Soil humidity sensors indirectly measure the volumetric water content using other soil properties, such as electrical resistance or dielectric constant. The moisture content of the soil can be determined by its effect on the dielectric constant by measuring the capacity between two electrodes implanted in the soil. The probe reading is not linear with the water content and is influenced by soil type and soil temperature, so it requires perfect calibration.
- The sensor for reading the concentration of carbon monoxide, MQ7, is based on a sensitive material, this being SnO2, which has a lower conductivity in the "clean" air. As the concentration of carbon monoxide (CO) increases, the conductivity of the sensor increases. The sensor detects carbon monoxide at concentrations between 10 and 10,000 ppm.

The manner in which all these sensors are connected on the Arduino UNO board is shown in Fig. 1.

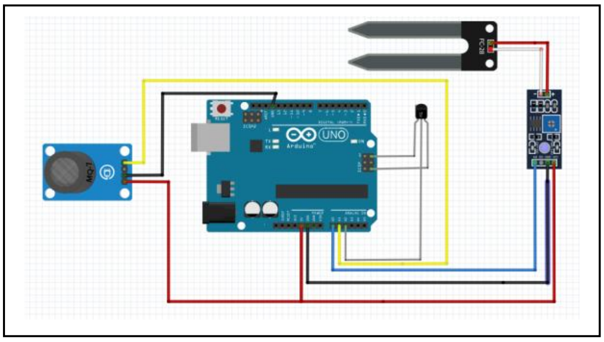

*Fig. 1 Graphical diagram of the circuit.*

## III. SOFTWARE IMPLEMENTATION

In the following section, the software implementation of the various components is discussed and the problems that had to be solved.

### *A. The Arduino Application*

Arduino offers an Integrated Development Environment (IDE), which is a cross-platform application written in Java. A program written in IDE for Arduino is called a sketch. Arduino IDE supports C and C ++ programming languages using special code organization rules.

- setup():a function that is run only once at the beginning of the program, when the settings are initialized.
- loop():a function repeatedly called until the power supply of the board is switched off.

For creating the source code, the following algorithm was used. To program the sensors on the Arduino microcontroller, their related analog pins should be initialized inside program variables:

```
int umiditate = A0; // soil humidity pin
int monoxid = A1;// CO concentration pin
float temperature = A2; // temperature pin
```

Next step is for configuring the direction of the analog pins, as these can be defined as inputs or outputs. For the purpose of sensor reading, these must be defined as inputs:

```
pinMode(temperatura,INPUT);
pinMode(monoxid,INPUT);
pinMode(umiditate,INPUT);
```

In the infinite loop, the program must read the values from the sensors using the analogRead() method. To correctly display the temperature value, one must first convert the number of 10 bit-precision to a voltage reading and multiply it by 1000 to obtain the pin voltage in mV. This number must then be divided by 10, because each increase of a Celsius degree in temperature corresponds to a 10 millivolts increase in voltage, according to the datasheet of the LM35 sensor [6]:

```
void loop(){
//reading of CO level
int masurare_monoxid = analogRead(monoxid);
//reading of soil humidity
int val_umiditate = analogRead(umiditate);
//temperature reading
int temp = analogRead(temperatura);
float val_temperatura = (5*temp*100/1024.0);
```

### *B. LabVIEW Virtual Instrument*

LabVIEW is mainly used in the construction of virtual instruments, having specialized functions on remote acquisition, processing, display and transmission of measurement signals, but can also be regarded as a general-purpose graphic programming language. LabVIEW is a very flexible programming environment, with powerful libraries of dedicated functions. Thus, it is possible to design and develop applications from the simplest, such as measuring and displaying a temperature, to sophisticated online testing and industrial control programs [7].

The first step was the instrument's front panel design (Fig. 2). As it can be seen in this figure, the front panel contains an interface for every sensor explained in the previous section. After building the front panel, the code was added in the Block diagram (Fig. 3) using graphical representations of functions to control the front panel objects. Front panel objects appear as terminals on the block diagram.

To connect the Arduino microcontroller to the virtual instrument, Linx tool was installed from the MarkerHub tools as the preferred method of choice.

At its core, LINX is a hardware abstraction layer that allows you to connect to a single LabVIEW interface to a variety of different hardware devices. Depending on the device, it can be locally accessed in one of two ways: Input / Output (I / O). Another alternative for connecting the Arduino microcontroller to LabVIEW is the NI-VISA driver, created by those at National Instruments, which allows the implementation of the I / O standard. The choice of the LINX driver, created by Digilent, was made for this specific project beacuse it is free and can be downloaded from the official website of the National Instruments.

To load LINX Device Firmware, one must follow the steps:

- Click Tools , next MakerHub, next LINX, and in the final Firmware Wizard.
- Select Device Family and Device Type.
- Select the COM Port associated device;
- Select the Firmware Version.

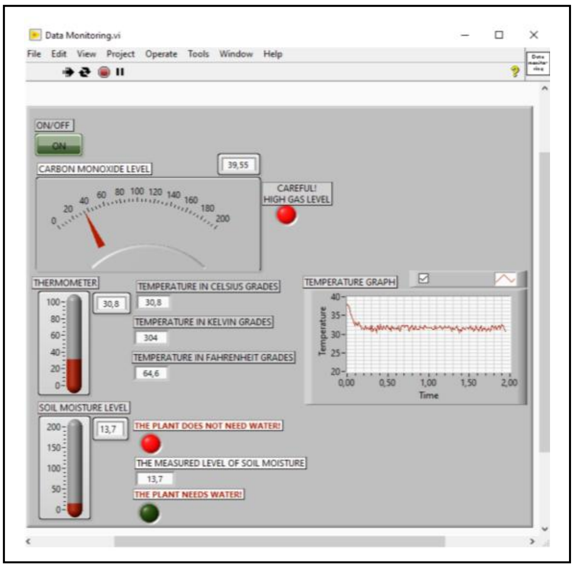

*Fig. 2 Front panel of the virtual instrument.*

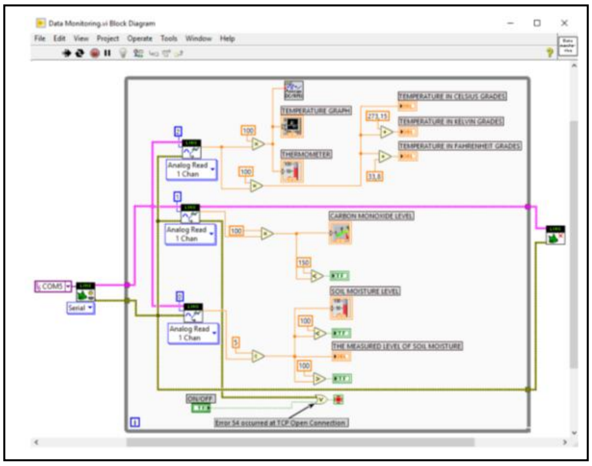

*Fig. 3 – Block diagram of the virtual instrument.*

## IV. CONCLUSIONS

In conclusion, being able to visualize and analyze the data from the sensors, one can create automatic control systems. Upon reaching critical levels (temperature, gas, etc.) these systems have the possibility to intervene, ensuring our protection and comfort.

The functioning of this device was presented in the national SCCSS (Student Scientific Communications Sessions) in 2018, which was held in Târgoviște [8].

As a future improvement to this system, the authors would like to add a GSM module for transmitting information to a phone in real time and alert the user in case of an eminent danger. Another Wireless module would transmit the data towards a web server where data statistics will be stored and processed.

## V. REFERENCES

[1] IOPSCIENCE, Real-time virtual instrumentation of Arduino and LabVIEW based PV panel characteristics, 1 January, 2019, web, https://iopscience.iop.org/article/10.1088/1755-1315/161/1/012019/, Accessed: 10/12/2019.

[2] Cepișcă C., Jula N., “Traductoare și senzori”, București, Editura ICPE, 1998.

[3] Chiță Monica-A., Pantelimon B., Șchiopu P., Iliescu C., “Senzori și traductoare-Curs”, Pitești, Univ. Pitești, 2001.

[4] Electrical4U, Characteristics of Sensors, 30 December, 2018, web, https://www.electrical4u.com/characteristics-of-sensors/, Accessed: 10/12/2019.

[5] Miron M, Miron L, “Măsurări Electrice și Electronice”, Brașov, Editura Academiei Forțelor Aeriene “Henri Coandă”, 2003.

[6] TEXAS INSTRUMENTS, LM35 Precision Centigrade Temperature Sensors, 30 September, 2019, web, http://www.ti.com/lit/ds/symlink/lm35.pdf, Accesed: 10/12/2019.

[7] Teodorescu Rodica-Mihaela, Liţă Ioan, Cioc Ion Bogdan, Vişan Daniel Alexandru, Radu Dragoş-Gheorghe, “Tehnici pentru generarea şi vizualizarea virtuală a semnalelor”, ISBN 978-606-560-289-2, nr. pag. 142, Editura Universităţii din Piteşti, http://biblioteca.upit.ro/, 2012.

[8] SESIUNEA DE COMUNICARI A CERCURILOR STIINTFICE STUDENTEŞTI IN DOMENIILE – INGINERIE ELECTRICĂ, ELECTRONICĂ CONTROL ŞI CALCULATOARE – SCCSS – IEECC, TARGOVISTE 12- 13 IULIE 2018, 30 Aprilie, 2019, web, http://www.sccss.ro/2018/index.html, Accessed: 10/12/2019.