Annals of the University of Petrosani, Electrical Engineering, 27 (2025)

A VIRTUAL INSTRUMENT FOR TEMPERATURE MONITORING THROUGH THE USE OF INTEGRATED SENSORS

NICOLAE PĂTRĂȘCOIU[1], VALI – CHIVUȚA SÎRB[2]

Abstract: The automatic or autonomous control of a process involves the monitoring of its parameters and their regulation in order to adapt their values relative to those predefined by the user. In this paper, the control of the environment in which the process takes place is assumed, and the monitored parameter is the temperature of this environment. Temperature values are acquired using an integrated temperature transducer from the LM9402x series, whose operation is based on the effect of temperature variations on the electrical characteristics of the p–n junction. In order to ensure the autonomy of the monitoring system with respect to the process control system, a hardware structure variant based on the widely adopted open-source Arduino platform is proposed. Although the programming software associated with this platform is based on a standard C++ compiler, in this work LabVIEW programming is employed so as to enable integration into a hierarchical virtual instrument for monitoring the entire process.

Key words: temperature, integrated sensors, Arduino, LabVIEW.

1. GENERAL CONSIDERATIONS ON TEMPERATURE MEASUREMENT

In order to measure temperatures, it is necessary to define a precise scale with stable and reproducible values, between which interpolation relationships can be established, and which should be as close as possible to the thermodynamic temperature scale derived from the laws of thermodynamics. Currently, the International Practical Temperature Scale of 1968 (IPTS-68) is used, which establishes that the unit of temperature is the Kelvin [K], equal to 1/273.16 of the thermodynamic temperature corresponding to the triple point of water.

In practice, tolerated units of measurement are also used, namely: the degree Celsius (t [°C]), the degree Fahrenheit (t [°F]), and the degree Rankine (t [°R]) [1].

The principles used in the construction of temperature transducers are: resistive, thermoelectric, semiconductor conduction, optical, or piezoelectric.

[1] *Ph.D., Associate Prof. Eng.,University of Petroşani, nicolaepatrascoiu@upet.ro*
[2] *Ph.D., Lecturer Eng., University of Petroşani, sirb_vali@yahoo.com*

Regardless of the principle used, a thermal contact between the sensing element and the environment or object being measured is required. Such a contact is considered perfect when there is no longer a thermal gradient between the sensing element and the medium or object. Achieving such contact requires a certain amount of time necessary to reach thermal equilibrium, which is a slow process [2], [3], [4].

The amount of heat exchanged per unit time between the sensing element, at an instantaneous temperature *T*, and the measured environment, is given by:

$$dQ{=}a \cdot A \cdot \left(T_1 \text{-} T\right) \cdot dt \tag{1}$$

where *a* represents the thermal conductivity and *A* the contact surface area.

Considering the specific heat *c* of the sensing element and its mass *m*, the amount of heat absorbed by it is:

$$dQ = m \cdot c \cdot dT \tag{2}$$

Assuming that heat losses during transfer are negligible, based on relations (1) and (1), one may write:

$$a \cdot A \cdot \left(T_1 - T\right) \cdot dt = m \cdot c \cdot dt \tag{3}$$

Introducing the thermal time constant, $\tau_T = \left(m \cdot c\right)/\left(a \cdot A\right)$ the solution of the differential equation (3) is obtained as:

$$T = T_1 - k \cdot e^{-\frac{t}{\tau_T}} \tag{4}$$

The graph of the temperature response based on solution (4) is presented in fig.1.a. In obtaining this response, the amount of heat released or absorbed by the sensing element—through which it modifies the initial measured temperature from T_1 to T_2— was not taken into account. Thus, the measured medium or object was approximated by an ideal model represented by an infinite heat source.

Taking into account the influence of the sensing element on the measured quantity, represented by the temperature variation *ΔT*, in practice one does not measure the actual temperature T_1, but its modified value T_2 (fig.1.b).

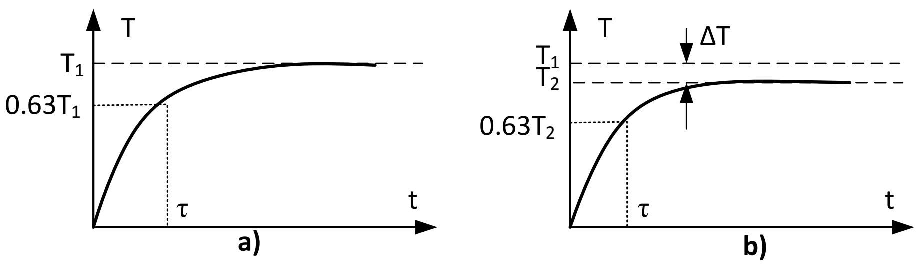

Fig.1. Temperature response of the sensor

Theoretically, an infinite time is required to achieve thermal equilibrium between the sensing element and the environment (object), but for sufficiently good accuracy this time may be considered to be 0.5–1 min [5], [6], [7].

2. INTEGRATED TEMPERATURE SENSOR LM94021

The LM94021 is a CMOS-based integrated temperature sensor that incorporates both the sensing element and the signal conditioning circuitry, providing a precise analog output voltage dependent on the temperature of the device package.

The sensor operates over a wide temperature range from −50 °C to +150 °C and generates an output voltage inversely proportional to the measured temperature. It is powered by a single-supply voltage starting at 1.5 V. Due to its low power consumption, the LM94021 is well suited for battery-powered systems as well as for general-purpose temperature monitoring applications.

The sensitivity of the sensor can be configured via two logic inputs, Gain Select 1 (GS1) and Gain Select 0 (GS0). These inputs allow the selection of the voltage-to-temperature transfer ratio, as summarized in Table 1.

Table 1. **Sensitivity configuration of the LM94021 sensor**

GS1	GS0	Sensitivity (mV/°C)
0	0	−5.5
0	1	−8.2
1	0	−10.9
1	1	−13.6

The gain selection inputs are compatible with TTL logic levels, enabling dynamic gain optimization during operation or facilitating system diagnostics.

The transfer characteristics of the LM94021 sensor are illustrated in Figure 2.a, while the recommended application circuit is shown in Figure 2.b.

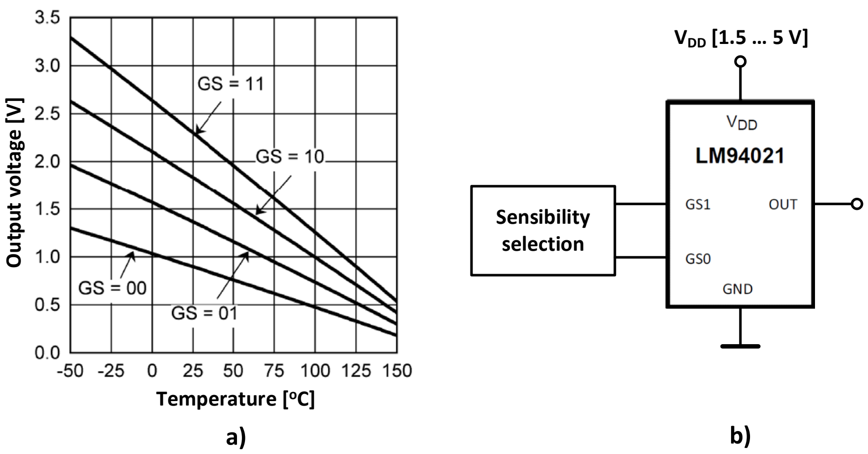

Fig.2. Integrated temperature sensor LM94021

The electrical characteristics corresponding to supply voltages in the range of 1.5 V to 5.5 V are provided in Table 2.

Table 2. **Sensitivity configuration of the LM94021 sensor**

PARAMETER		CONDITIONS	TYPICAL (1)	LIMITS (2)	UNITS (LIMIT)
	Sensor Gain	GS1 = 0, GS0 = 0 GS1 = 0, GS1 = 1 GS1 = 1, GS0 = 0 GS1 = 1, GS0 = 1	−5.5 −8.2 −10.9 −13.6		mV/°C mV/°C mV/°C mV/°C
	Load Regulation (3)	Source ≤ 2.0 μA (4) Sink ≤ 100 μA Sink = 50 μA	 0.4	**−1** **1.6**	mV (max) mV (max) mV
	Line Regulation (5)	$(V_{DD} - V_{OUT}) \geq 200$ mV	200		μV/V
I_S	Supply Current	T_A = +30°C to +150°C T_A = −50°C to +150°C	9	**12** **13**	μA (max) μA (max)
C_L	Output Load Capacitance		1100		pF (max)
	Power-on Time (6)	C_L= 0 pF C_L=1100 pF	0.7 0.8	**1.6** **2.4**	ms (max) ms (max)
V_{IH}	GS1 and GS0 Input Logic "1" Threshold Voltage			**V_{DD}- 0.5V**	V (min)
V_{IL}	GS1 and GS0 Input Logic "0" Threshold Voltage			**0.5**	V (max)
I_{IH}	Logic "1" Input Current (7)		0.001	1	μA (max)
I_{IL}	Logic "0" Input Current (7)		0.001	1	μA (max)

Although the LM94021 exhibits good linearity over its operating range, its response is not perfectly linear, displaying a slightly parabolic behavior. Nevertheless, the transfer characteristics can be considered approximately linear for most practical applications.

For higher accuracy, the manufacturer provides tabulated output voltage values (in millivolts) for all four selectable sensitivities across the entire temperature range. These data can be implemented using a look-up table approach in sensor-based applications. The complete transfer tables are available from the manufacturer's official documentation.

The values in this table can also be determined analytically based on the expressions defined for the 4 sensitivities as follows:

1. For GS1=0 and GS2=0

$$V_{TEMP}\left[mV\right]=870,6mV-5,506\frac{mV}{^oC}\cdot\left(T-30^oC\right)-0,00176\frac{mV}{^oC}\cdot\left(T-30^oC\right)^2 \quad (5)$$

2. For GS1=0 and GS2=1

$$V_{TEMP}\left[mV\right]=1324,0mV-8,194\frac{mV}{^oC}\cdot\left(T-30^oC\right)-0,00262\frac{mV}{^oC}\cdot\left(T-30^oC\right)^2 \quad (6)$$

3. For GS1=1 and GS2=0

$$V_{TEMP}\left[mV\right]=1777,3mV-10,888\frac{mV}{^oC}\cdot\left(T-30^oC\right)-0,00347\frac{mV}{^oC}\cdot\left(T-30^oC\right)^2 \quad (7)$$

4. For GS1=1 and GS2=1

$$V_{TEMP}\left[mV\right]=2230{,}8mV-13{,}582\frac{mV}{^{o}C}\cdot\left(T-30^{o}C\right)-0{,}00433\frac{mV}{^{o}C}\cdot\left(T-30^{o}C\right)^{2} \quad (8)$$

Additionally, the transfer characteristics may be analytically approximated using sensitivity-specific expressions corresponding to each GS1/GS0 configuration. For linear approximation within a limited temperature interval, two reference points are selected from the transfer table, and the sensor response is modeled using a first-order linear equation.

Let (T_1, V_1) and (T_2, V_2) denote the lower and upper reference points, respectively. The linear approximation between these points is defined by a standard interpolation relationship, represented by (9).

$$V-V_1=\left(\frac{V_2-V_1}{T_2-T_1}\right)\cdot\left(T-T_1\right) \quad (9)$$

where

- ✓ T_1 [°C] and V_1 [mV] are the coordinates of the lower point;
- ✓ T_2 [°C] and V_2 [mV] are the coordinates of the upper point.

3. IMPLEMENTATION OF THE TEMPERATURE MONITORING SYSTEM

3.1. Hardware Structure

For temperature monitoring, the LM94021 sensor manufactured by Texas Instruments was tested. The sensor is mounted on the LM9402xEVM Evaluation Module (EVM), produced by the same manufacturer.

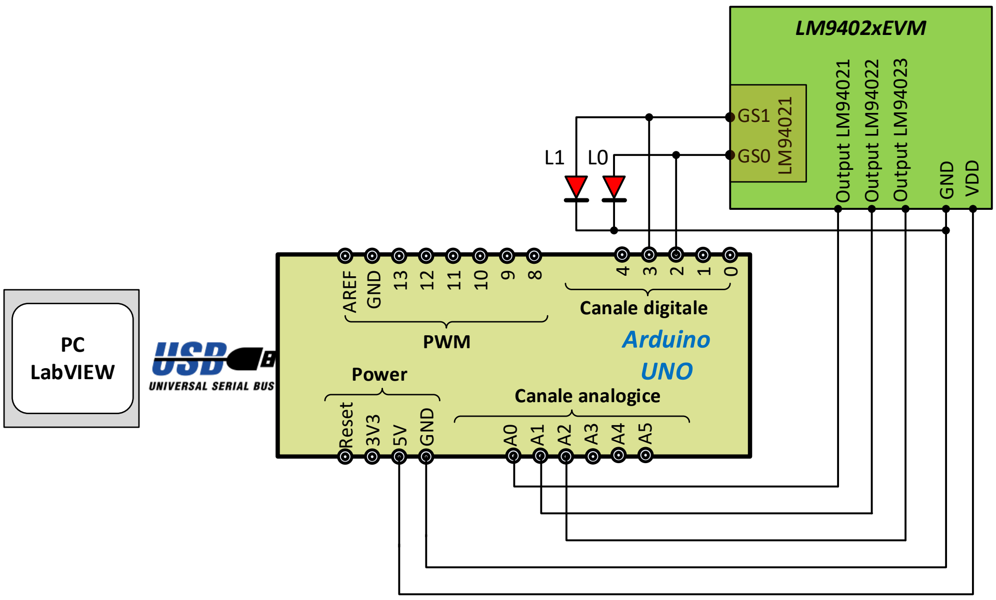

Fig.3. Block diagram of the monitoring system

Fig. 3 presents the block diagram of the monitoring system.

The temperature information, represented by the voltage at the sensor output, is acquired using the Arduino Uno module, specifically through the analog input channel AI0, to which the sensor signal is connected. Other analog quantities can be acquired using the remaining analog channels by appropriate selection.

The Arduino Uno module also performs a first stage of signal processing, namely the analog-to-digital conversion, using the converter integrated into the ATmega328 microcontroller on the Arduino Uno board.

Communication between a computer and the Arduino Uno module is achieved via the USB serial interface. Thus, the acquired data can be retrieved and further processed using an application developed in LabVIEW, which provides the information to the user in a synthetic and accessible form through an appropriate graphical interface. Through the same interface, the user can also generate commands or control signals for the monitored technical system.

Sensor sensitivity control is performed via two digital channels, allowing the user to select any of the four available sensitivity levels through the application. Two LEDs are associated with these channels and display the selected sensitivity in binary format. The use of these LEDs also demonstrates the capability of the user to generate commands or controls for the monitored technical system.

3.2. Hardware Structure

The Texas Instruments LM9402xEVM evaluation module (EVM), shown in Fig. 4, is designed to evaluate the operation and performance of the LM94021, LM94022, and LM94023 sensors [8].

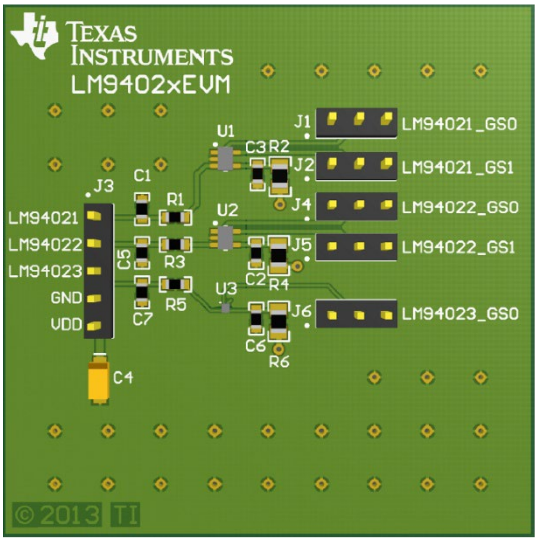

Fig.4. Texas Instruments LM9402xEVM evaluation module

The LM94021, LM94022, and LM94023 devices are precision analog-output temperature sensors implemented in integrated CMOS technology. They feature sensitivity selection pins that allow the user to configure the output sensitivity of the temperature–voltage transfer characteristic [9], [10].

The LM9402xEVM module includes all three analog temperature sensors, as well as the necessary jumpers for sensitivity configuration, either by connection to the supply voltage VDD (GSx = 1) or to ground GND (GSx = 0).

Sensitivity control can also be achieved using appropriate logic levels provided by the digital channels of the Arduino Uno module and applied to the same jumper terminals.

The jumpers have the following functions:

- ✓ J1 – Sensitivity selection terminal (GS0) for LM94021. The 0.1-inch header allows the user to set the transfer function sensitivity to HIGH (GS0 = 1) or LOW (GS0 = 0).
- ✓ J2 – Sensitivity selection terminal (GS1) for LM94021. The 0.1-inch header allows the user to set the transfer function sensitivity to HIGH (GS1 = 1) or LOW (GS1 = 0).
- ✓ J4 – Sensitivity selection terminal (GS0) for LM94022. The 0.1-inch header allows the user to set the transfer function sensitivity to HIGH (GS0 = 1) or LOW (GS0 = 0).
- ✓ J5 – Sensitivity selection terminal (GS1) for LM94022. The 0.1-inch header allows the user to set the transfer function sensitivity to HIGH (GS1 = 1) or LOW (GS1 = 0).
- ✓ J6 – Sensitivity selection terminal (GS0) for LM94023. The 0.1-inch header allows the user to set the transfer function sensitivity to HIGH (GS0 = 1) or LOW (GS0 = 0).
  - J3.P1 – Analog temperature sensor output for LM94021.
  - J3.P2 – Analog temperature sensor output for LM94022.
  - J3.P3 – Analog temperature sensor output for LM94023.
  - J3.P5 (GND) – Ground reference pin.
  - J3.P6 (VDD) – Supply voltage for LM94021, LM94022, and LM94023.

The schematic diagram of the Texas Instruments LM9402xEVM evaluation module, highlighting the components used, is shown in Fig. 5.

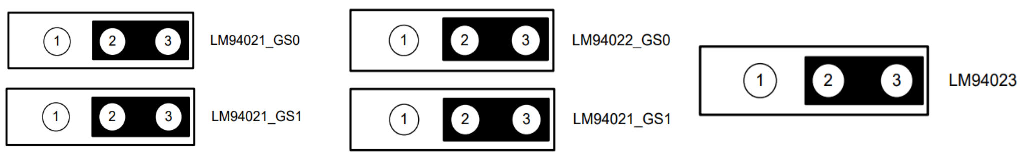

Fig.5. Default configuration of jumpers

3.3. Virtual Instrument

The virtual instrument is a program developed in LabVIEW that manages communication with the Arduino Uno module and performs data processing for the information acquired from the sensors via the Arduino module [11], [12].

3.3.1. Front Panel of the Virtual Instrument

The front panel contains the user interface elements, which can be used for:

- ✓ data or command input (input-type elements), referred to as controls;
- ✓ display of results obtained through data processing, referred to as indicators.

Figure 6. shows screenshots of the front panel corresponding to the operation of the virtual instrument.

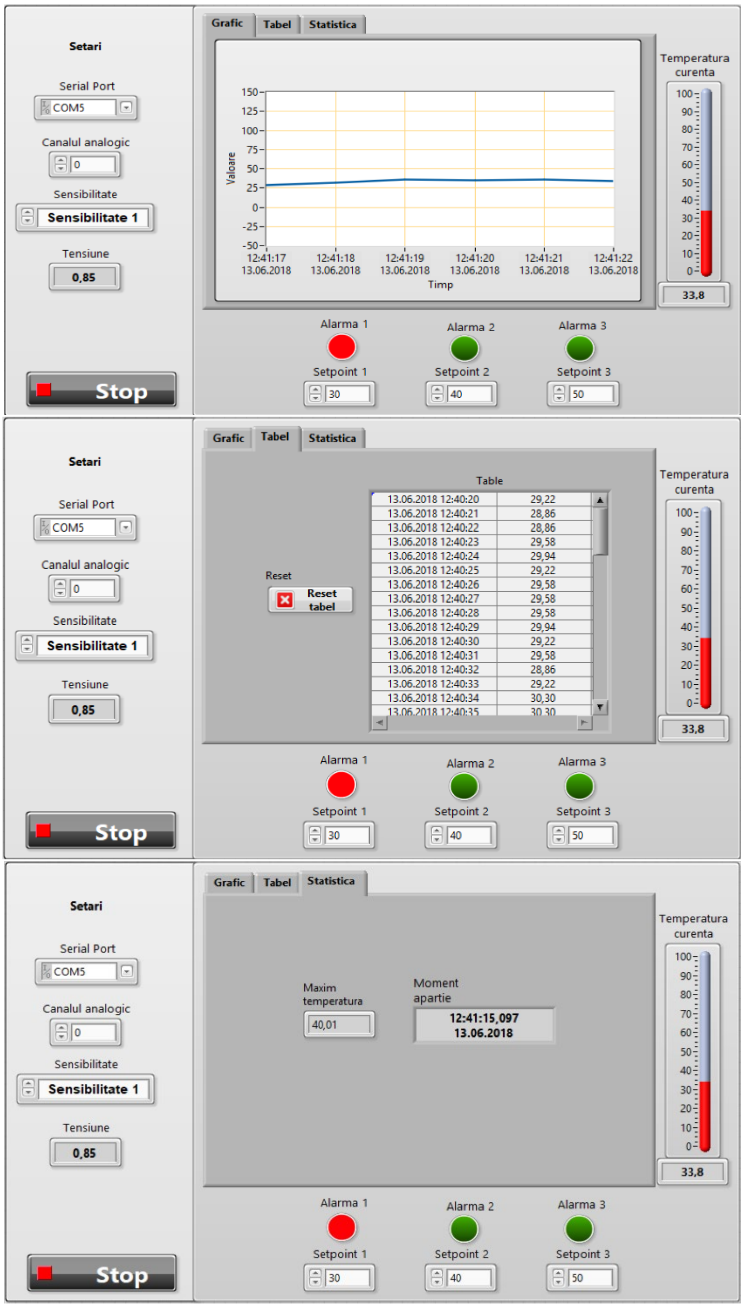

Fig.6. Front panel of the virtual instrument

In the implementation of this application, three component windows of the front panel are used, the user having the possibility to choose any of them at any time. These three windows with the names **Graph**, **Table**, **Statistics** can be selected via the Tab command.

The **Graph** window is used to display the time evolution of the monitored temperature values. The time units on the horizontal axis represent real time, including date, hour, minute, and second corresponding to each sensor reading.

The **Table** window provides a tabular representation of the temperature values and their corresponding acquisition times.

The **Statistics** window displays the maximum identified temperature value over the entire acquisition period, together with the time at which it was detected.

The front panel also includes the following indicators:

- ✓ **Voltage** – numeric indicator displaying the sensor output voltage in volts;
- ✓ **Current Temperature** – slider-type indicator displaying the current temperature value in degrees;
- ✓ **Alarm 1**, **Alarm 2**, **Alarm 3** – Boolean indicators signaling the exceedance of the configured alarm thresholds.

The front panel controls and their defined functions are:

- ✓ **Serial Port** – ring-type control used to select the USB port to which the Arduino Uno module is connected;
- ✓ **Analog Channel** – numeric control used to select the analog input to which the sensor is connected;
- ✓ **Sensitivity** – ring-type control used to select one of the four available sensor sensitivity levels;
- ✓ **Setpoint 1**, **Setpoint 2**, **Setpoint 3** – numeric controls used to set the alarm thresholds;
- ✓ **STOP** – Boolean control used to stop program execution.

3.3.2. Block Diagram of the Virtual Instrument

The block diagram represents the program itself and consists of operations, functions, and programming structures arranged according to the data flow required for execution.

Program execution continues until the *STOP* button on the front panel is activated. This behavior is implemented by enclosing all operations and functions within a *While* structure. A delay of 1 second between consecutive cycles is introduced using the *Wait Until Next ms.vi* function, resulting in a temperature acquisition rate of one sample per second.

The *Open Serial.vi* function, which establishes serial communication with the module controlled via the *LINX* package (in this case, the Arduino Uno), and the *Close.vi* function, which terminates communication and releases system resources, are excluded from the *While loop*. These functions are executed at program start and upon activation of the *STOP button*, respectively.

LINX is a utility developed by National Instruments that enables the integration of Arduino development boards into the LabVIEW programming environment, without requiring direct embedded programming of the microcontroller. Communication between LabVIEW and the Arduino platform is typically achieved via a USB serial interface, based on a client–server architecture in which the Arduino runs *LINX* firmware, while the LabVIEW application controls system operation [14].

Through *LINX*, users can access the analog and digital input/output resources of the Arduino board, as well as certain communication interfaces, by means of dedicated Virtual Instruments (VIs) available in LabVIEW. This approach facilitates the rapid development of data acquisition, monitoring, and control applications in a graphical and intuitive manner.

The overall block diagram of the virtual instrument is shown in Figure 7.

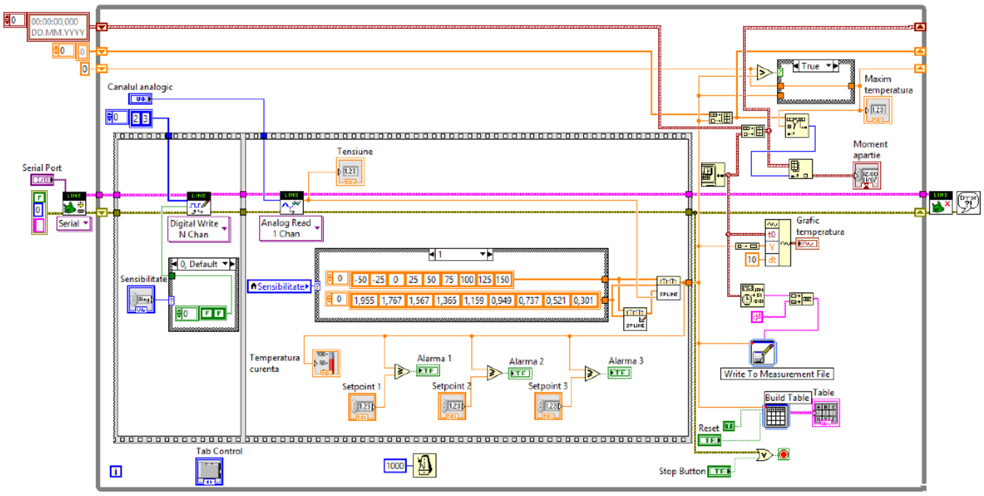

Fig.7. Block diagram of the virtual instrument

Data acquisition and processing are performed in two stages:

1. Selection of the sensitivity value;
2. Reading the sensor voltage and performing the voltage-to-temperature conversion.

This functionality is implemented using a *Sequence* structure and the *Digital Write.vi* function, which writes a Boolean value to a selected digital output, and the *Analog Read.vi* function, which reads the value available at the selected analog input. Digital outputs DIO2 and DIO3 are used to control GS0 and GS1, respectively, while the analog input is selected via the **Analog Channel control**.

The **Sensitivity control** selects one of the four required combinations according to Table 1, generated using a *CASE* structure, as shown in Figure 8.

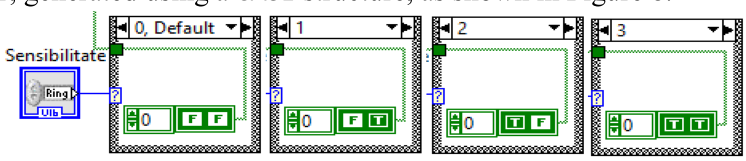

Fig.8. Sensitivity selection

For voltage-to-temperature conversion, several approaches are possible, including the use of analytical relations or lookup tables. However, these methods either require complex implementation or significant memory allocation. To avoid these limitations while ensuring validity over the entire measurement range, an interpolation-based method was selected.

Spline interpolation is applied using a limited set of voltage–temperature pairs from the sensor datasheet for each sensitivity configuration. The Spline Interpolant.vi and *Spline Interpolation.vi* functions are used, as illustrated in Figure 9.

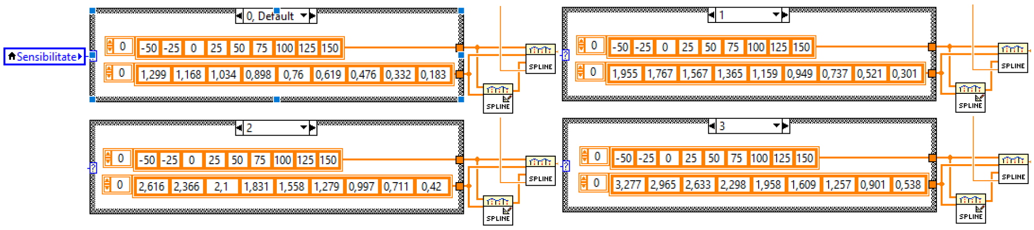

Fig. 9. Voltage-to-temperature conversion

The *Spline Interpolation.vi* function returns the interpolated temperature value corresponding to the measured voltage, based on the coefficients generated by the *Spline Interpolant.vi* function, which computes the second derivative of the spline interpolation function at the specified data points.

The obtained temperature values are compared with the alarm thresholds set using the **Setpoint 1** to **Setpoint 3** controls. When a threshold is exceeded, the corresponding LED indicators **Alarm 1** to **Alarm 3** are activated, as shown in Figure 10.

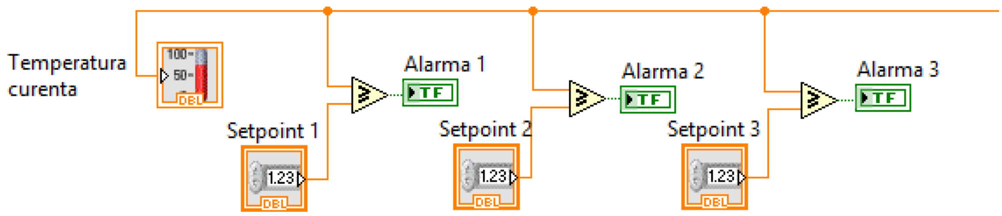

Fig.10. Alarm generation

The identification of the maximum temperature value during acquisition, as well as the time at which it occurs, is performed in each acquisition cycle using a Shift Register. This mechanism allows the comparison of the previously identified maximum with the newly acquired value and retention of the higher value.

Simultaneously, temperature values and their corresponding timestamps are stored and indexed in vectors. Using the index of the maximum temperature value, the associated acquisition time can be determined. Figure 11 highlights the portion of the virtual instrument responsible for identifying the maximum temperature value and its corresponding time.

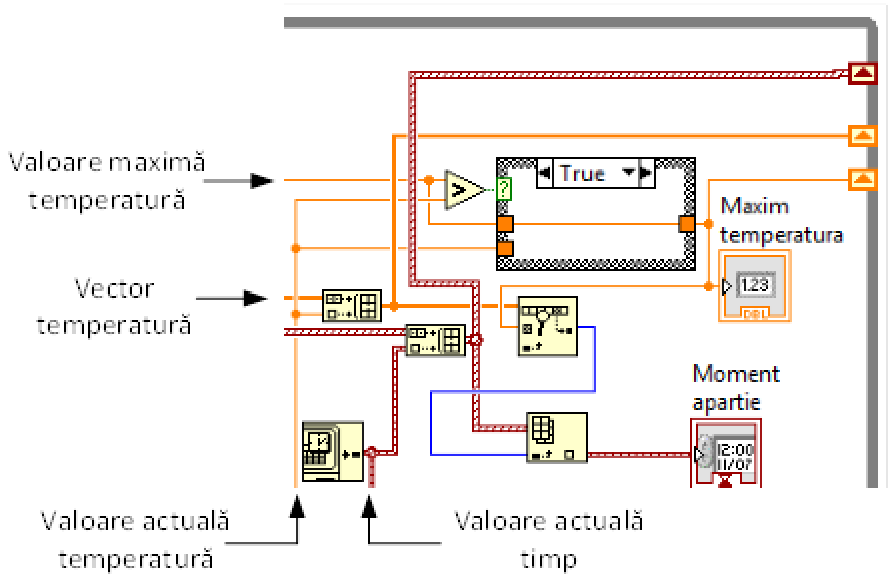

Fig. 11. Identification of the maximum temperature value

The acquired temperature values are exported in *.lvm* format to a file that can subsequently be opened in a Microsoft Excel spreadsheet and used to maintain a historical record of the measured quantities. For this purpose, the *Write to Measurement File Express VI* function is employed, as illustrated in Figure 12.

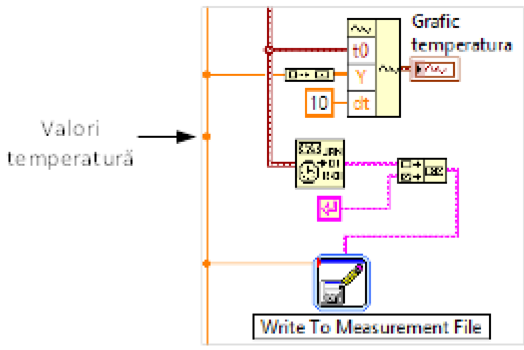

Fig. 12. Export of the acquired values

This function enables the writing of data to text-based measurement files (*.lvm*), binary measurement files (*.tdm* or *.tdms*), or Microsoft Excel files (*.xlsx*).

4. CONCLUSIONS

Temperature monitoring is important not only for obtaining information about its value, but also because many physical processes are influenced by temperature. Based on temperature-related information, it is therefore possible to derive relevant insights and to monitor and control such physical processes.

In temperature evaluation through measurement, the selection of the measuring instrument, and particularly of the sensor, is of critical importance. The sensor must reach thermal equilibrium with the surrounding environment as rapidly as possible, a process that depends on the thermal transfer properties between the environment and the sensor. Under certain conditions, the temperature of the sensor itself may generate a temperature gradient, leading to discrepancies between the measured temperature and the actual temperature of the system. This gradient increases with the mass of the sensor. In such cases, the measured temperature varies not only with the system temperature but also with the thermal transfer characteristics of the system.

The above conditions are best satisfied by sensors based on the properties of the p–n junction, as employed in the LM9402x series of temperature sensors manufactured by Texas Instruments and used in the present application.

Arduino represents a platform through which information systems capable of "perceiving" the surrounding environment via sensors and "controlling" it through actuators can be developed. This platform is open-source and consists of both a development environment and a development board based on an AVR microcontroller, which can be interfaced using virtual instrumentation applications.

Virtual instruments are employed to ensure flexibility and enhanced data processing capabilities for both software and hardware systems, particularly the personal computer, in applications designed to develop, test, or control processes and systems while performing precise measurements of both analog and digital signals. Moreover, they enable the creation of user-defined systems tailored to specific application requirements. Industries with automated processes widely employ virtual instrumentation systems to improve productivity, reliability, safety, optimization, and overall system stability.

REFERENCES

[1]. Hillken K. D., Steele A. G., *The International Temperature Scale: Past, Present, and Future*, NCSLI Measure J. Meas. Sci., Vol. 9, No. 1, pp. 60-67, 2014.

[2]. Bentley J. P., *Principles of Measurement Systems*, Pearson Education, London, 2005.

[3]. Doebelin E. O., Manik D. N., *Measurement Systems: Application and Design*, McGraw-Hill, New York, 2011.

[4]. Fraden J., *Handbook of Modern Sensors: Physics, Designs, and Applications*, 5th edition, Springer, New York, 2019.

[5]. Morris A. S., Langari R., *Measurement and Instrumentation: Theory and Application*, 2nd edition, Academic Press (Elsevier), 2020.

[6]. Webster J. G., **Eren H.**, *Measurement, Instrumentation, and Sensors Handbook*, 2nd edition, CRC Press, Boca Raton, 2018.

[7]. Hymczak H. *Core Temperature Measurement — Principles of Correct Measurement, Problems and Complications*. International Journal of Environmental Research and Public Health (MDPI), 18(20), 10606, 2021.

[8]. Blum J., *Exploring Arduino: Tools and Techniques for Engineering Wizardry* Wilye, 2015.

[9]. Evans W. B., *Arduino Programming Notebook*, Creative Common, 2017.

[10]. **Schwartz M., Manickum O.**, *Programming Arduino with LabVIEW*, Packt Publishing Ltd., Birmingham, 2015.

[11]. ******* *LabVIEW User Manual*; 2025 Edition; National Instruments Corp. Austin, Texas, U.S.A., https://docs-be.ni.com/bundle/labview/preprocessedpdf/enus

[12]. ***** Texas Instruments**, *LM9402XEVM Evaluation Module (EVM). User's Guide*, SNIU017, 2013.

[13]. ***** Texas Instruments**, *LM94021/LM94021Q Multi-Gain Analog Temperature Sensor*, Datasheet, 2013.

[14]. http://www.ti.com/lsds/ti/analog/temperature_sensor.page.