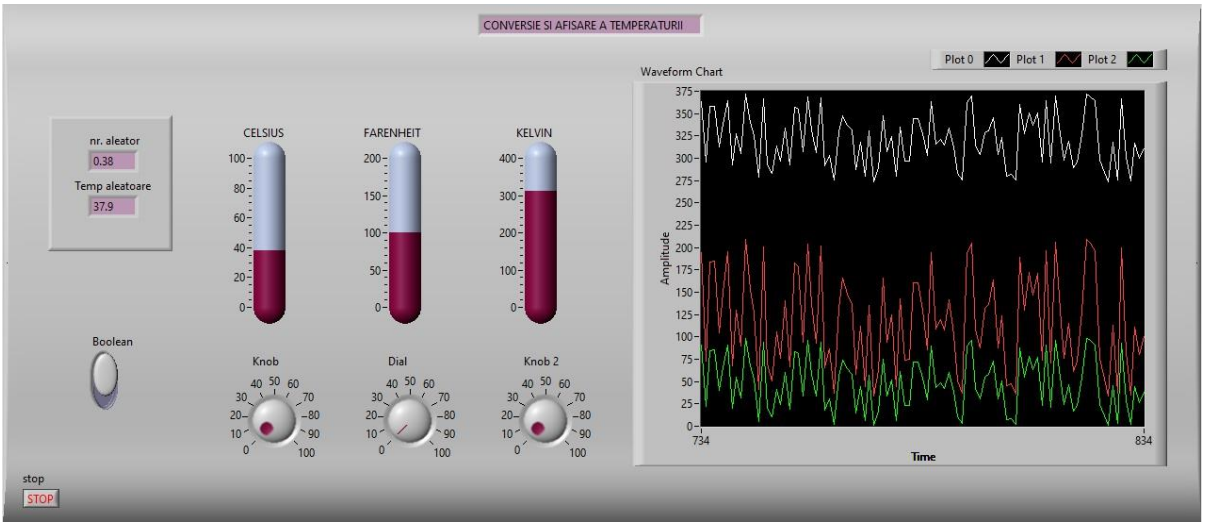
PROJECT 1
TEMPERATURE CONVERSION AND DISPLAY

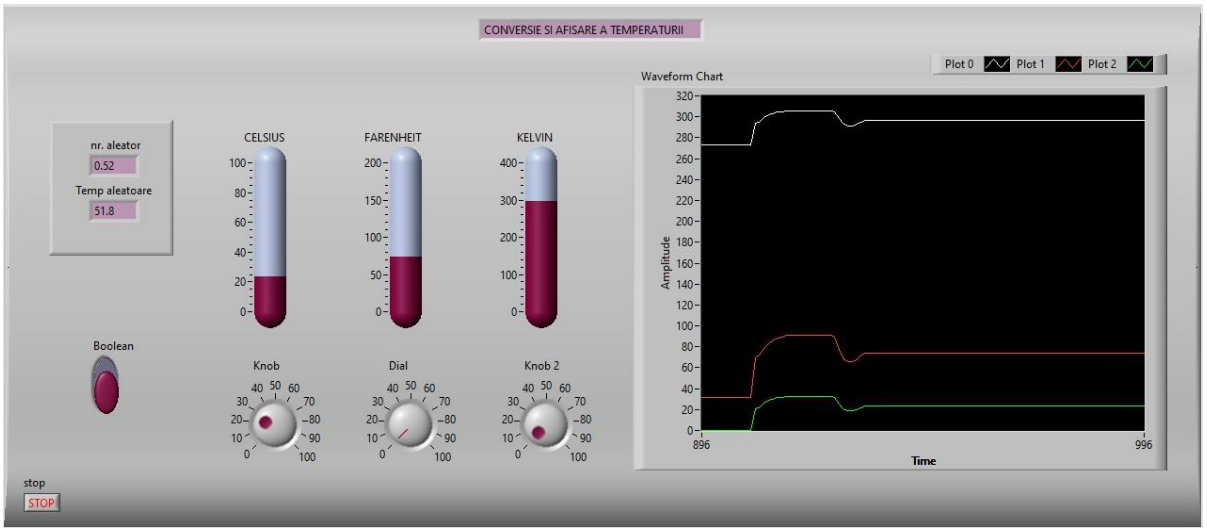
DESCRIPTION

The purpose of this application is to perform the conversion and graphical display of the temperature in degrees Celsius, Fahrenheit and Kelvin.

The application uses both random values for a more realistic simulation of temperatures and imposed values, provided by a controller. The choice of these types of values is based on a switch. Next, the three temperatures are displayed graphically.

FRONT PANEL



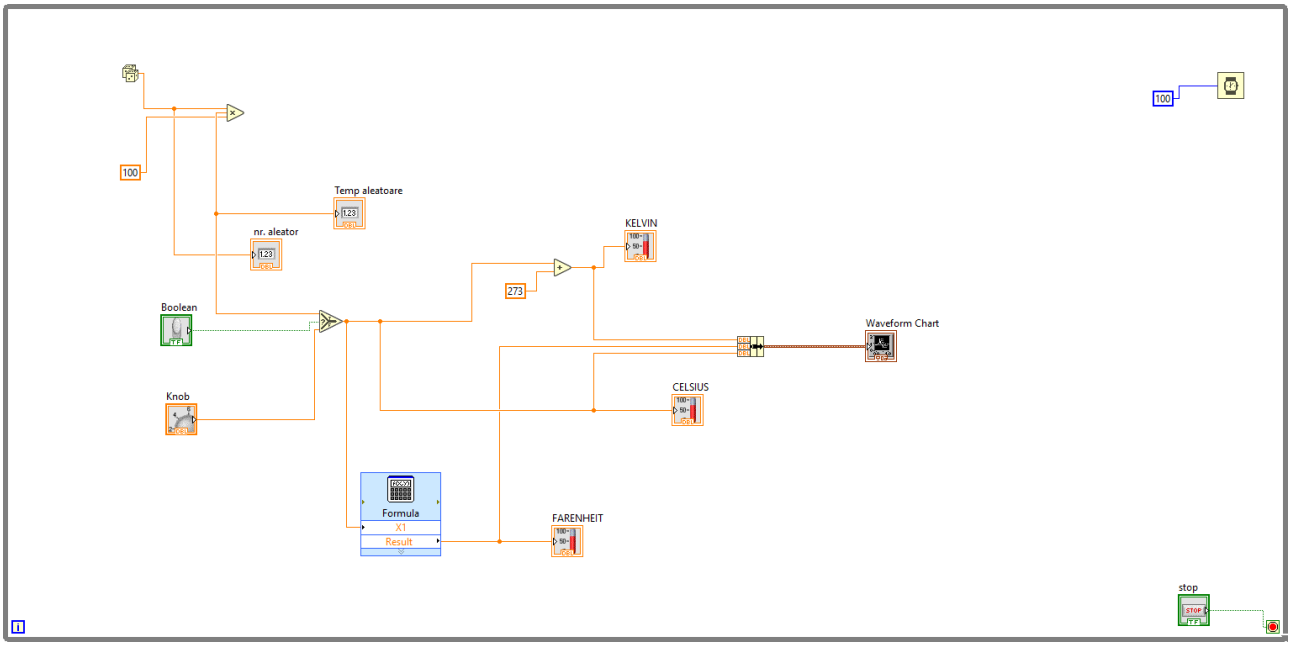


The application contains as follows:

1. Three thermometers with the role of indicator for displaying the converted temperature.
2. Three potentiometers to choose a higher or lower temperature.
3. A button for choosing the type of processed values.
4. A graphic display that displays the graph of the temperature function over time.
5. Three string pointers for displaying random values, scaled random values, and

plain text.

BLOCK DIAGRAM



In the block diagram of the application, you can find some components of the control panel mentioned above, such as the knobs for changing the temperature, the thermometers, the graph, the switch and the string indicators.

For rendering random values, I used the random function, which I multiplied by 100 to display values that exceed the range [0,1] that are characteristic of it.

To choose the type of processed values I used the select function and for the conversion from degrees Celsius to Fahrenheit I used a block in which I entered the appropriate formula.

Displaying the three temperatures on the same graphic is possible thanks to the bundle used, which is the link between the three values offered after conversion and the graphic display.

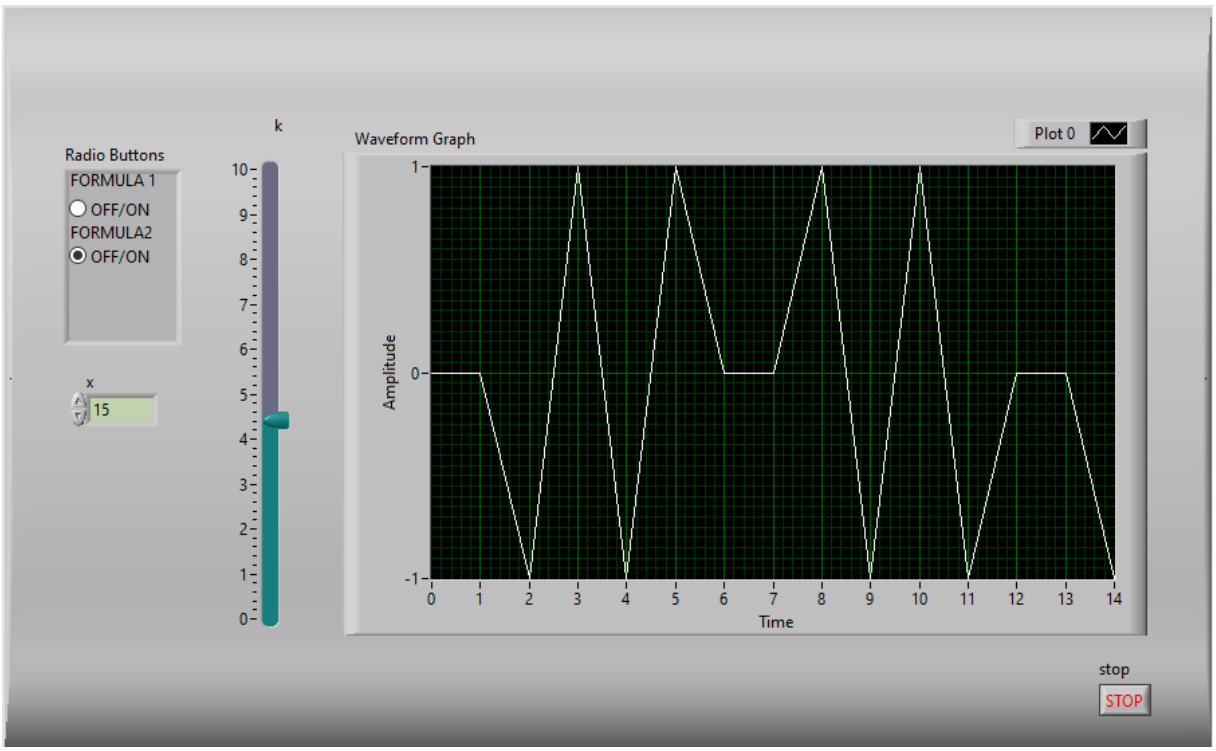
This whole scheme is contained in a while loop to which I added a timer of 100 for easy visualization of the temperature graphs.

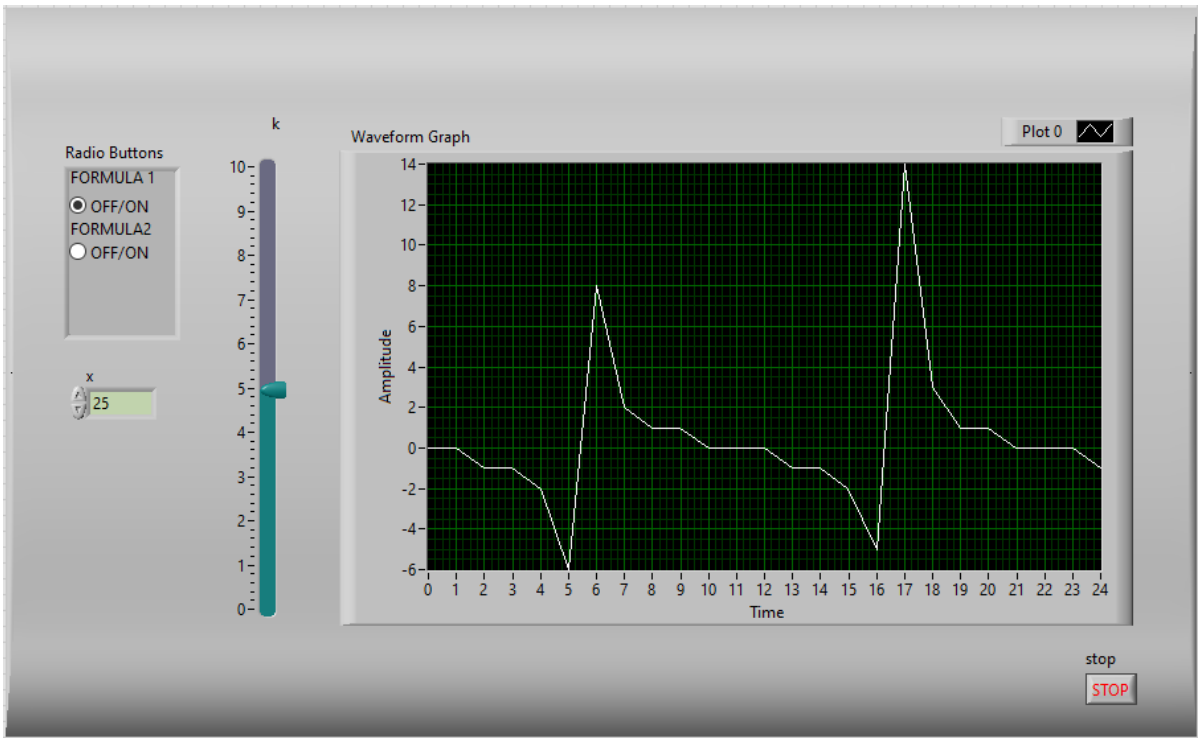
PROJECT 2
GRAPHIC DISPLAY OF A FUNCTION

DESCRIPTION

The present work allows the graphic realization of one or more functions depending on the variable k. The application allows both the choice of the variable k and the equation of the function that will be displayed graphically.

FRONT PANEL





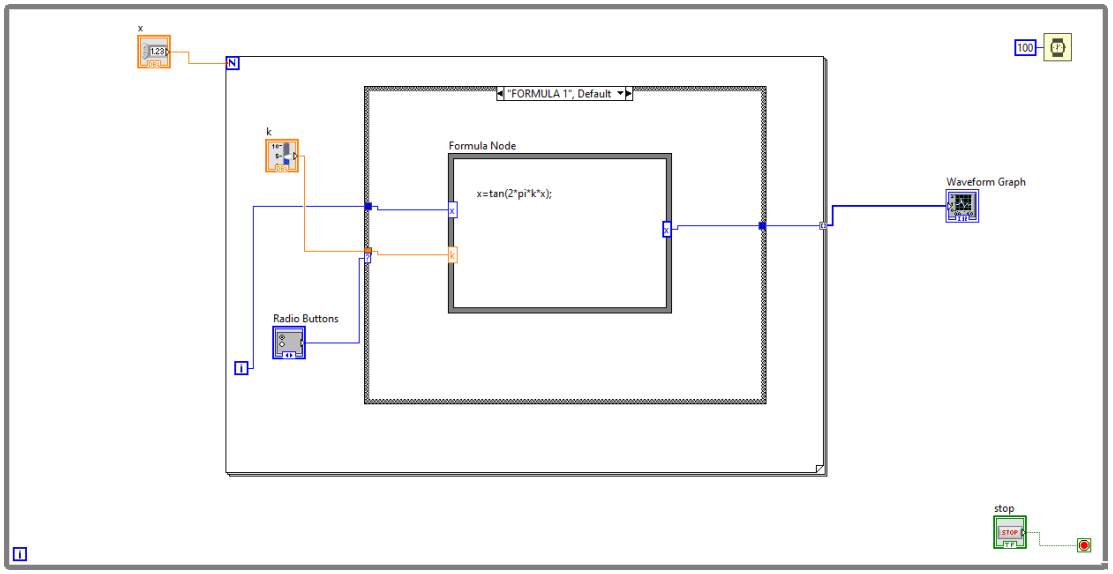
The application interface contains:

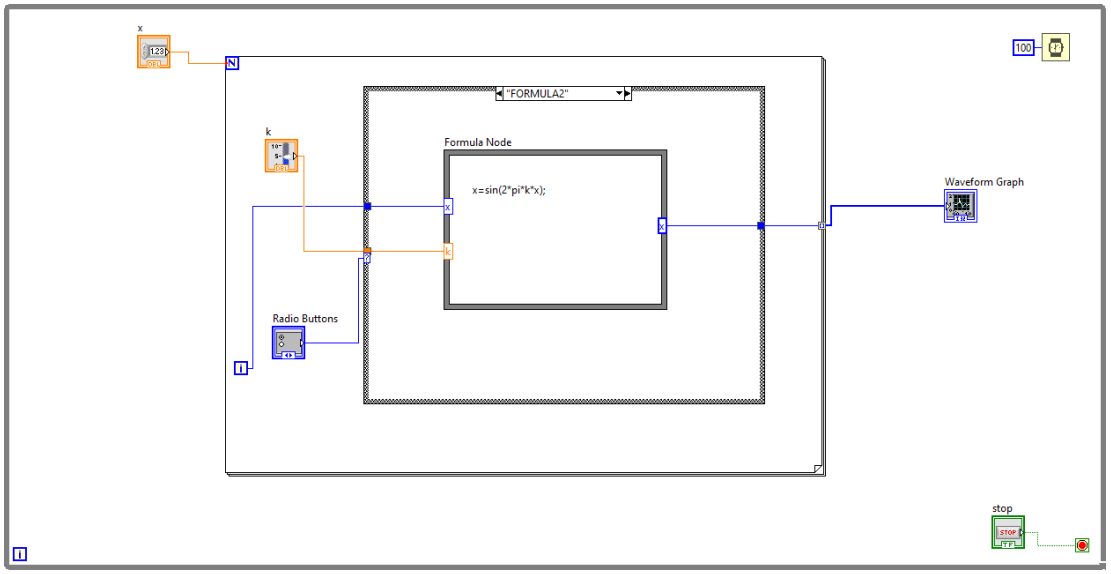
• A graph, waveform graph, which renders the graphic representation of the chosen function and which is modeled in real time with the change of the chosen values of the variable k.

• A slide-type control through which values are given to the variable k.

• A radio button that allows choosing the equation that needs to be displayed.

• A numeric control that represents the number of iterations chosen.





The block diagram of the program contains a while structure in which the following are composed:

• A timer with a constant of 100 milliseconds that allows the correct visualization of the generated function graphs.

• The waveform chart where the functions of the equations rendered from the output of the node formula are displayed.

• A numerical control that establishes the number of iterations to be executed, this being wired to the for loop structure through N.

• The stop button that stops the development of the application.

The while loop structure also contains a for loop structure in which you can find the slide for controlling the variable k and the radio button for choosing the desired equation. They are wired to a house structure through which this choice is

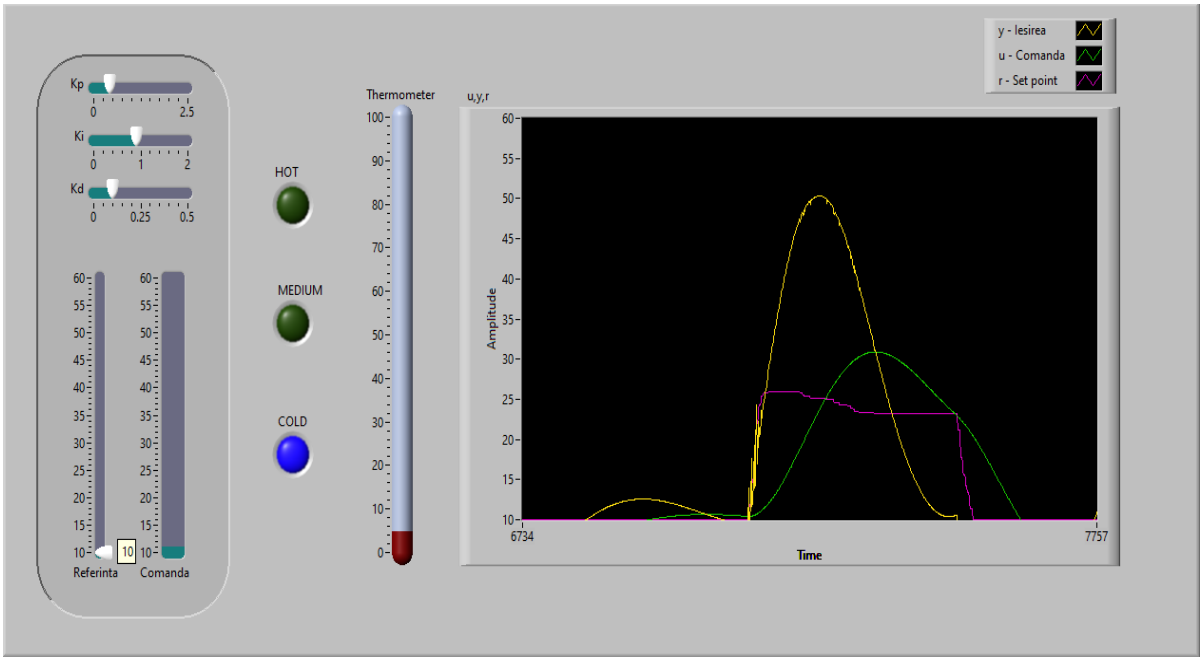
made. The description of the equations is thus made with the help of a node formula introduced in this case structure.

PROJECT 3
TEMPERATURE REGULATION

DESCRIPTION:

The present application adjusts the temperature in a system with the help of a PID type regulator.

FRONT PANEL



The front panel of this application consists of:

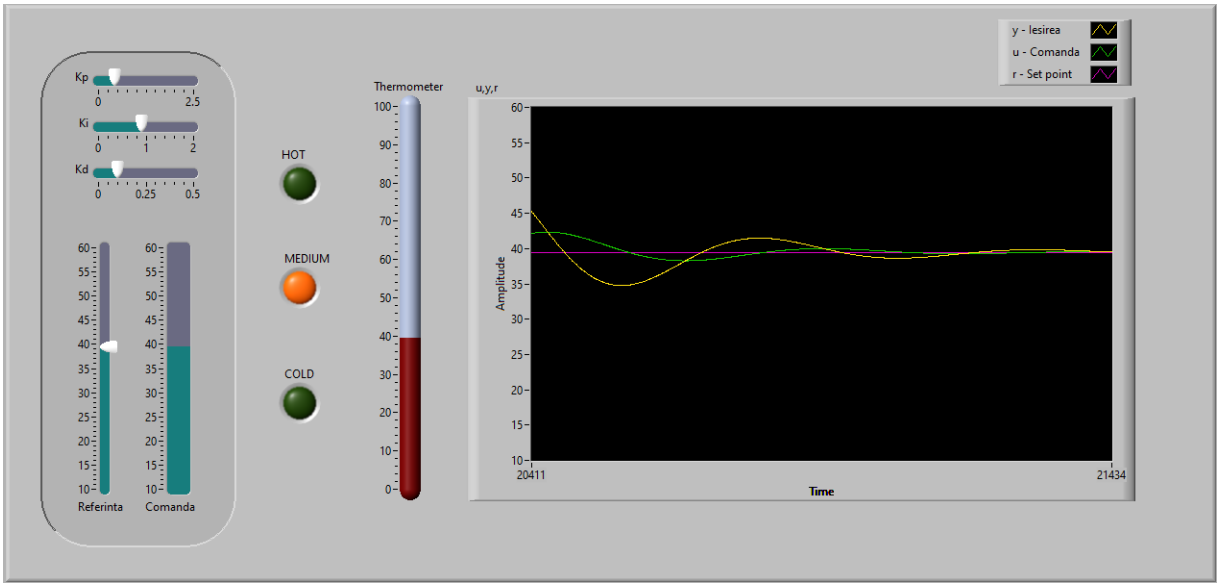
• Three slide controls that represent the parameters of the PID regulator, namely, kp, ki and kd.

• Another slide-type element through which the value of the system reference is controlled, but also one where the command that the regulator of the execution elements renders based on the change of the PID parameters and the chosen reference is represented.

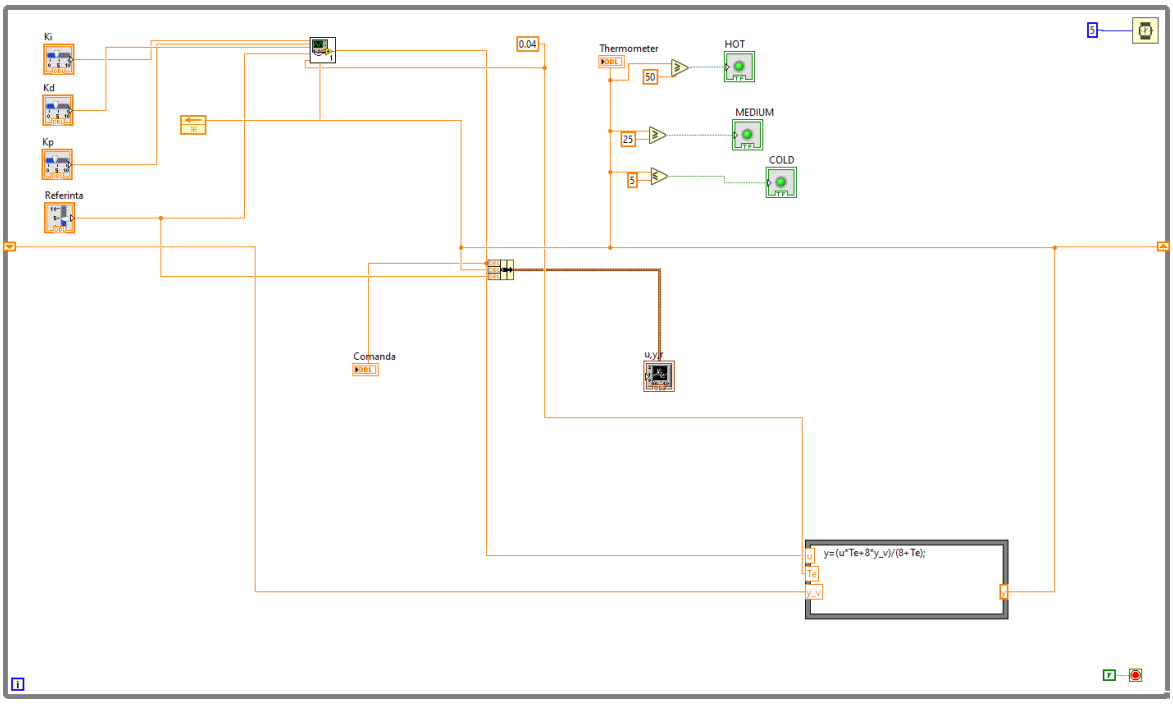
• A thermometer on which the value of the temperature we want to reach is represented.

• Three LEDs that highlight the temperature reached, defining it as warm, medium or hot.

• A graphic display that shows the response of the system after all the required parameters have been applied to it.



BLOCK DIAGRAM



The block diagram of the application is contained in a while loop that contains:

• Below the PID VI that greatly simplifies the block diagram is the schematic equivalent for creating the PID controller.

• Four slides that are the Sub VI PID inputs represented by the regulator parameters kp, ki, kd and the system reference.

• A formula block structure in which the characteristic equation of the process, in this case the temperature, is defined.

• A thermometer on which the temperature value is displayed and which is

connected to three LEDs which, according to certain conditions imposed by the comparison elements, become active or not.

• The regulator command that results from changing the initial parameters of the PID sub vi and which is also its output.

• A graph on which the output, the order, and the reference entered with the help of a bundle will be displayed on the same graph, allowing us to analyze them comparatively more easily.

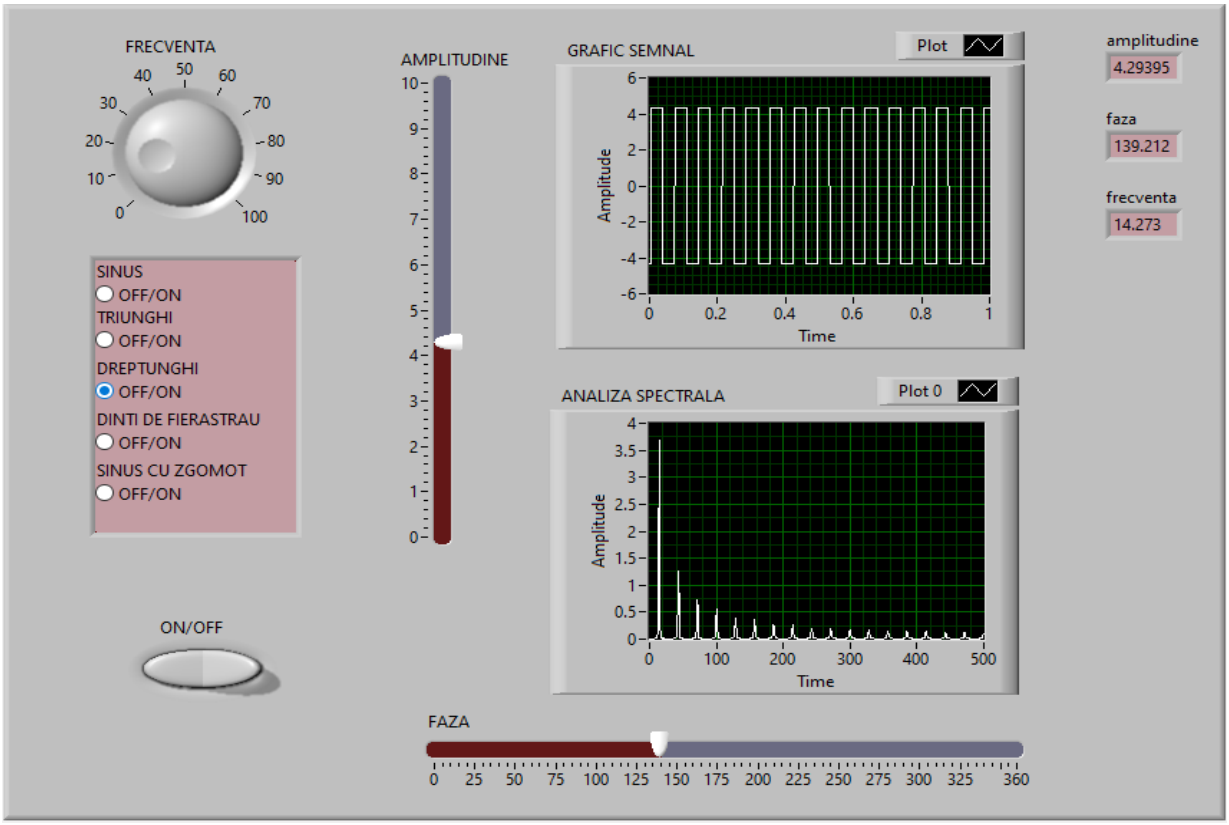
PROJECT 4

SIGNAL GENERATOR

DESCRIPTION

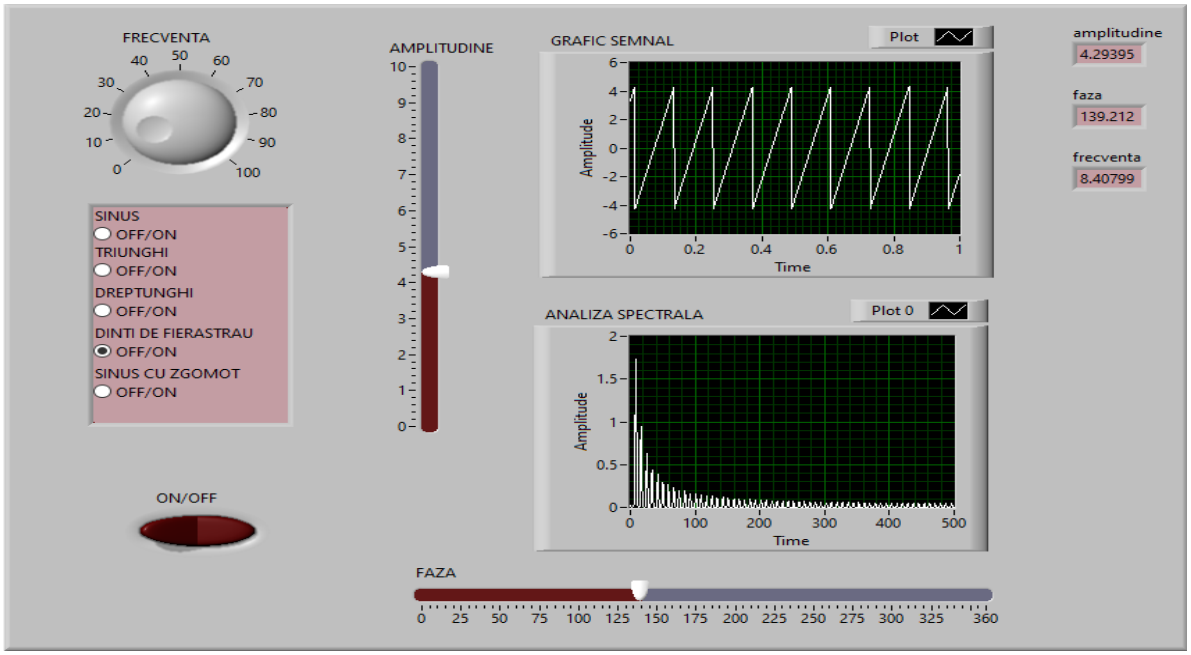
The present work represents a generator of different signals including sine or rectangle signal, at the user's choice, which offers the possibility of frequency, amplitude and phase control and which displays the signal flow and its spectral analysis.

FRONT PANEL



The application interface contains:

• Two graphs where the waveforms and spectral analysis are displayed for the chosen function which can be sine, triangle, rectangle, sawtooth or sine with noise.



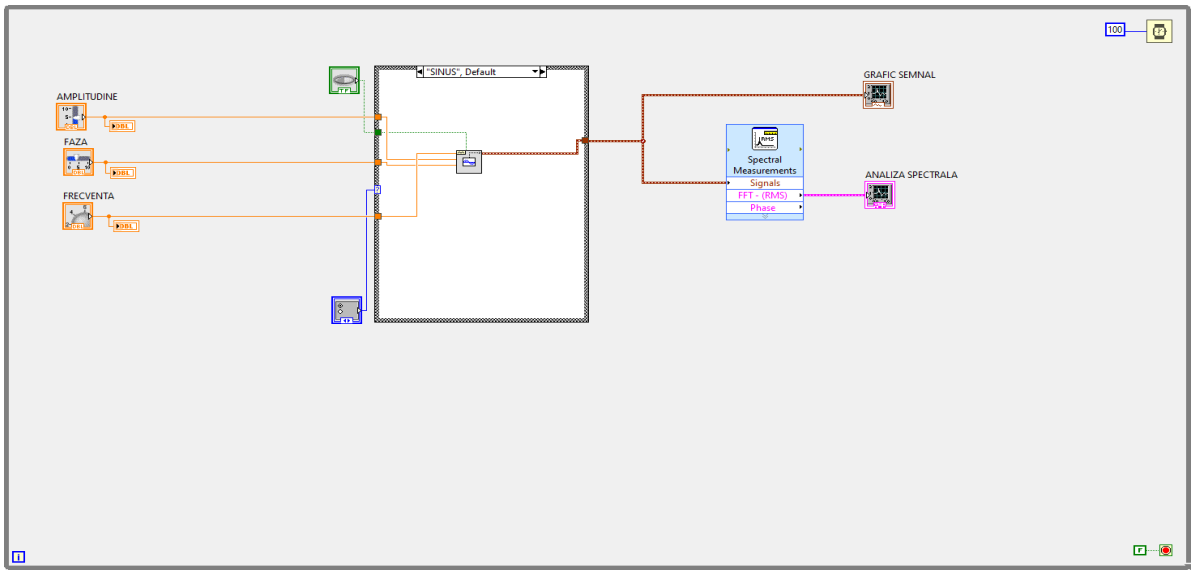
• Three knob and slide controllers that control the value of frequency, amplitude and phase.

• A switch on/off button that starts or stops the continuous playback of waveforms.

• A list of available signals for which we used a radio button.

• Three numerical indicators that provide the exact value of the value offered by the three controllers.

BLOCK DIAGRAM



The block diagram of the application contains a repetitive while loop structure that includes:

• A timer with a constant of 100 milliseconds that makes it easier to visualize the graphically displayed waveforms.

• A Radio button for selecting signals that is wired to a case selector in a case structure. Then, in this house structure, the various icons that represent these signals are added.

• Three controllers wired to three numeric indicators for accurate playback and display of amplitude, phase, and frequency values that are each wired to the input ports of the signal type. The output value from these signals is represented on the two graphs.

• The spectral analysis is done with the help of a spectral measurement found in the palette of waveform functions, namely in analog waveform measurements which is placed between the signal output and the graph representing the spectral analysis.

• To simulate the sinusoidal signal with noise, a generator found in the waveform to analog waveform generation function palette is used, to which the type of signal and the type of noise are set, in this case it is a uniform noise.