Heating and Ventilation System with Arduino and LabVIEW

Mihai Bogdan

Computer Science and Electrical Engineering Department,
Lucian Blaga University of Sibiu, 550025, Romania,
E-Mail: mihai.bogdan[at]ulbsibiu.ro

Abstract

This system aims to control the heating and ventilation system of the home, using the LabVIEW graphical programming environment. The system consists of a temperature sensor, 2 control circuits, and the Arduino mega 2560 platform, which will control the entire system. The first control circuit is used to start a fan for cooling the house, and the second control circuit is used to operate a relay of the boiler for heating the house. This system works like a thermostat, so it will keep the house at the set (desired) temperature.The temperature sensor measures the temperature inside the house and if the temperature is lower than the set one, Arduino activates the boiler relay to start heating the house. After reaching the set temperature, the heating of the house is stopped. The same thing happens if the temperature of the room is higher than the set temperature, Arduino controls the fan to cool the room.

Keywords: LM 35 DZ, Cooling Fan, Arduino, LabVIEW, LINX.

1 Introduction

Home automation ensures increased comfort through various functions offered by automation, such as: measuring and improving the air quality inside the home, adjusting the lighting according to the light level, or automating the heating system so that the temperature is always right for the user. A thermostat is a component of a control system which senses the temperature of a system so that the system's temperature is maintained near a desired set point. The thermostat does this by switching heating or cooling devices on or off or regulating the flow of a heat transfer fluid as needed, to maintain the correct temperature. A thermostat can often be the main control unit for a heating or cooling system, in applications ranging from ambient air control to automotive coolant control. Thermostats are used in any device or system that heats or cools to a setpoint temperature, examples include building heating, central heating, air conditioners, as well as kitchen equipment including ovens and refrigerators and medical and scientific incubators (https://en.wikipedia.org/wiki/ Thermostat).

The system for controlling the heating and ventilation of a room, contains a temperature sensor LM 35 DZ, a circuit for controlling the boiler relay and respectively a circuit for controlling the fan.

The LM35 series is an integrated circuit for the temperature measurement. Corresponding to each degree Celsius (°C) it gives 10 mV as output. It can sense temperature in the range of −55°C–+150°C.

The temperature sensor is connected to the A0analog input of the Arduino mega 2560. The control circuit of the ventilation system is connected to the 40-digital pin, of the Arduino mega 2560, and the control circuit of the boiler relay is connected to the 42digital pin number. To see if the boiler relay is activated, we mounted an LED to indicate this (Figure 1).

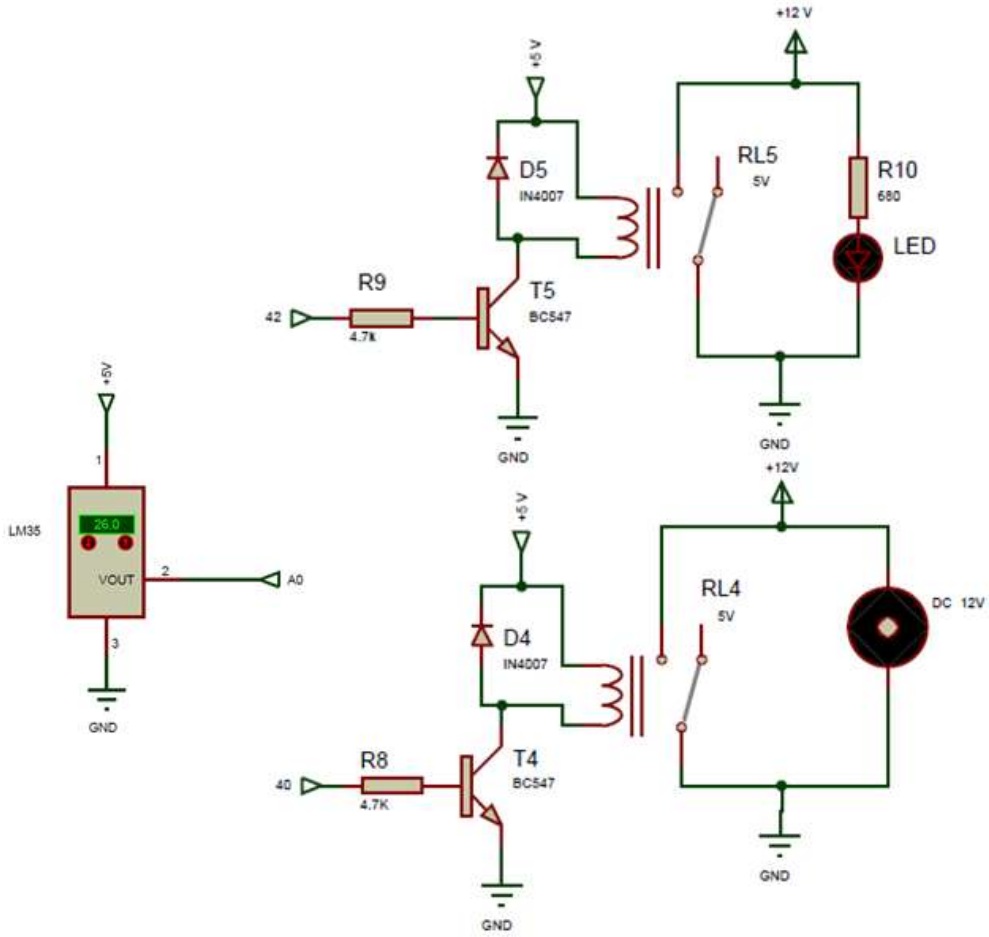


Figure 1.System Hardware Wiring Diagram

The resistance for limiting the current through the LED, we calculated it with the relation:

$$R_{LED} = \frac{U_{Al} - U_{LED}}{I_{LED}} = \frac{12 - 2}{15 \cdot 10^{-3}} = 666.6\Omega => R_{10} = 680\Omega$$

2 System Software Design

To create the interface between LabVIEW and Arduino, you need the following software:

- LabVIEW;
- NI VISA;
- VIPM;
- LINX

NI VISA - National Instruments Virtual Instrument Software Architecture is an API that provides a programming interface to control Ethernet/LXI, GPIB, serial, USB, PXI, and VXI instruments in National Instruments application development environments like LabVIEW, The API is installed through the NI-VISA driver (https://www.ni.com/ro-ro/support/documentation/supplemental/06/ni-visa-overview.html).

VIPM - VI Package Manager reduces project costs by helping you implement a code reuse process in your organization. VIPM makes it easy to manage and share reusable VIs across multiple projects, computers, and teams of developers (http://sine.ni.com/nips/cds/view/p/lang/ro/nid/209002.

LINX - LabVIEW for X (LINX) is designed to replace LabVIEW Interfaces for Arduino and to provide a LabVIEW generic protocol for the interface with any programmable device, but specifically targeting Microcontrollers and SoCs (System on a Chip). LINX will provide a high level of programming that allows users to communicate with several devices, including Arduino (https://www.labviewmakerhub.com/).

After installing all the necessary software, the LabVIEW program will open, in which I will make virtual instruments necessary for programming the Arduino MEGA 2560 boards, then select Tools >>Makerhub>> LINX >> LINX Firmware wizard.

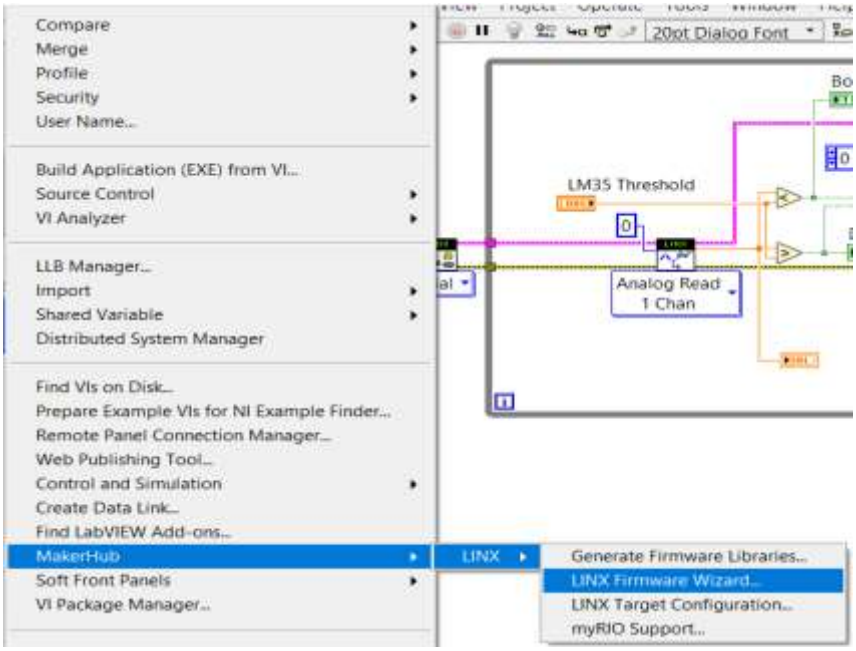


Figure 2.LINX Configuration

Represented in Figure 3 is the Front Panel of the VI. It contains the following controls and indicators:

- a control (Serial Port) for setting the serial port to which Arduino is connected;
- a boolean control (Stop Program) to stop running the virtual instrument;
- a numerical control (LM35 Threshold) for setting the voltage level, from which the relay of the boiler for heating the house or the fan for cooling the house is operated;
- a numerical indicator (LM35 Output Voltage) for displaying voltage values at the output of the LM35 temperature sensor;
-two Boolean indicators (Boiler Relay and DC Fan) to indicate the status of the boiler relay and DC fan.

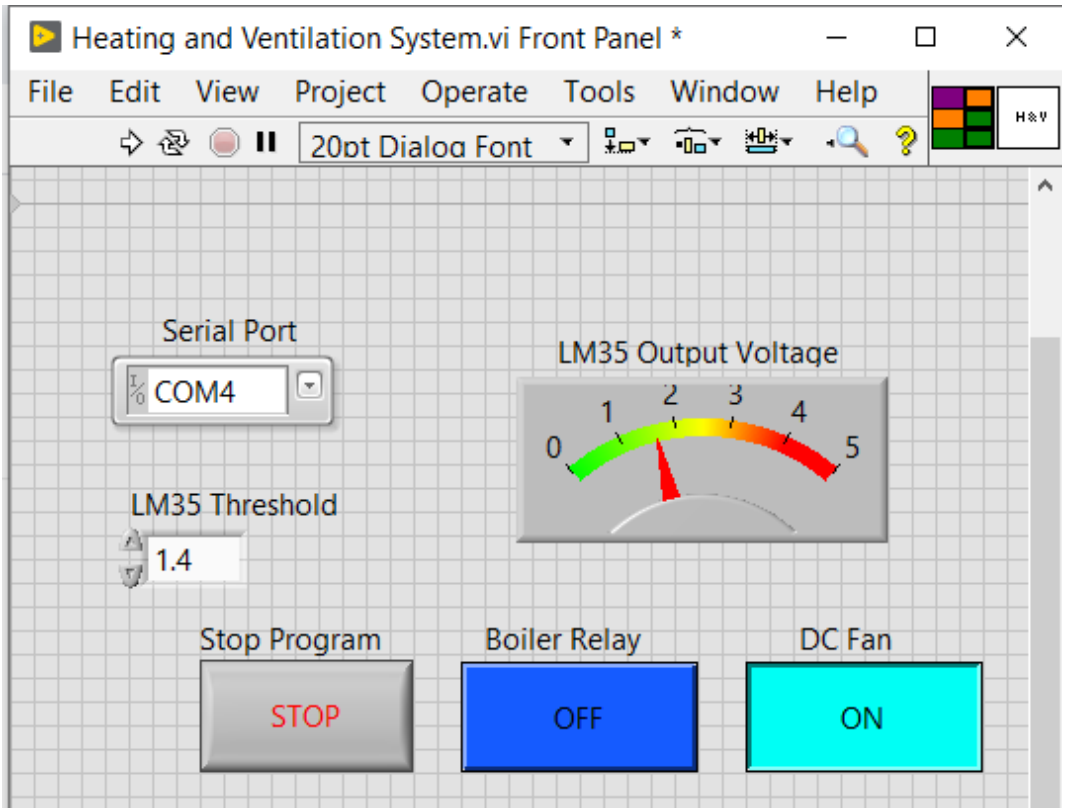


Figure 3. The Front Panel of the VI

The Block Diagram of VI contains the controls and indicators terminals of the Front Panel, the various nodes, constants, and the wires. The nodes in LabVIEW are different functions, subVIs, and programming structures (Bogdan, 2018).

In the block diagram, the following functions and programming structures were used:

- The **Open Serial** Function: this function opens a serial connection with the Arduino platform. Each different program starts with the Open function.

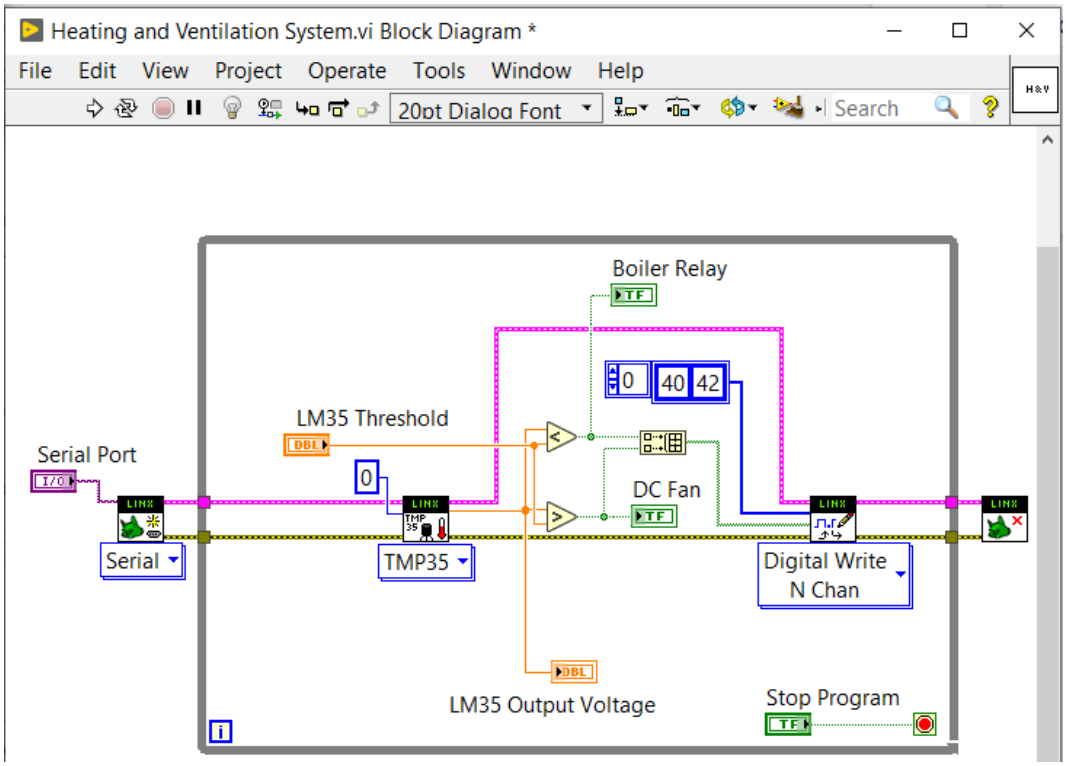
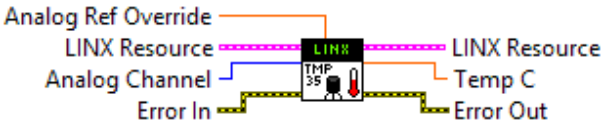


Figure 6. The Block Diagram of the VI

- The **TMP35** Function: Read temperature from LM35 sensor.



- The **Digital Write N Chan** Function: Write the digital values (high or low) to the specified digital output channel.

- The **Close** Function controls the end of the program. This function closes the connection to the remote LINX device and free any local I/O resources. We must finish each different program with Close function;

-  The **Build Array** Function - this function adds the two boolean values to the

input, in a two-element array, which will be applied to the input of the Digital Write N Chan Function which will write these values to pins 40 and 42.

-  The **While Loop** structure: that will continuously execute the functions inside it until the conditions for termination are reached.

3 Conclusion

The two major heating and ventilation functions for a home, offer thermal comfort and acceptable indoor air quality at reasonable installation, operation, and maintenance costs.

I opted to write the application code in LabVIEW Graphical Programming and not in a classic text-based programming language.

The advantages of programming using the LabVIEW graphical programming language over the classic text-based programming language are:

- It offers a multitude of libraries and virtual tools specific to programming embedded systems;
- The graphical interface of the LabVIEW program is very friendly with the programmer;
- Higher productivity of graphic language compared to classical languages of programming;
- The LabVIEW language features interactive debugging tools.

References

Bogdan, M., (2019): „Monitoring and Alarming System for a Gas Central Heating Boiler", Proceedings of the 14th International Conference on Virtual Learning (ICVL 2019), ISSN: 1844-8933 - ISI Proceedings, p. 325-331, OCTOBER 25-26, 2019

Bogdan M., (2018): „Gas Detector Using Arduino and LabVIEW", Proceedings of the 13th International Conference on Virtual Learning ICVL 2018, ISSN: 1844-8933, p. 315-318, October 26-27, 2018

https://playground.arduino.cc/Main/MQGas Sensors

https://en.wikipedia.org/wiki/ Thermostat

https://www.ni.com/ro-ro/support/documentation/supplemental/06/ni-visa-overview.html

http://sine.ni.com/nips/cds/view/p/lang/ro/nid/209002

https://www.labviewmakerhub.com/