Embedded System for Controlling the Temperature Inside of an Electric Furnace

Andreea-Maria Neacă , Mitică Iustinian Neacă
Department of Electrical, Energetic and Aerospace Engineering,Faculty of Electrical Engineering, Craiova, Romania
neaca_andreea@yahoo.com
Department of Electrical, Energetic and Aerospace Engineering, Faculty of Electrical Engineering, Craiova, Romania
ineaca@elth.ucv.ro

***Abstract*— We have been listening for a long time to the wide functionality that industrial automation technologies can offer for improving the processes and adding security to the plants and the parts of them. However the price and an unstable market have restricted the deployment of such systems. Nowadays, companies offer too technology-dependent solutions which do not cover user demands completely. Meanwhile, papers from the literature focus on small innovations on specific parts of the industrial automation systems, which do not consider integration and deployment issues, in order to present practical designs [1]. The system presented in this work considers user requirements, including novel advances, all in an integral automation solution suitable for thermal treatments. The modular nature of the architecture allows direct adaptation to specific cases using standard technologies, and a proposal of an IP-based network for connecting the main automation module with the rest of plant. The main purpose of the entire plant is to monitor and control in real-time the temperature of the parts inserted into a resistive furnace. The software is based on m-files that contain the source code that calculates the control coefficients for the next step for ventilation and heating, it verifies the failures that could occur and creates a matrix of the coefficients, in order to make graphics.**

I. INTRODUCTION

With rising costs of energy, slimmer profit margins and tighter specifications on product quality, the metals-processing industry continues to look for ways to increase productivity and reduce costs.

PID controllers are standard tools for industrial applications. They are used mostly because they are simple and robust. It is also a disadvantage concerning in the fact that it is difficult to define the PID controller parameters properly. Therefore auto-tuning methods are developed. These methods enable the control practical and accurately.

A multivariable control strategy can provide tremendous benefits to the brazing process. Current fixed operating conditions such as furnace temperature set points can be used as manipulated variables to maintain a reducing atmosphere in the furnace.

Considering the characteristics of the heat treatment electric furnace, such as large time-delay, time-varying parameters and nonlinearity, it is difficult to realize a good result with traditional PID (proportion-integral-derivative) control, a novel predictive functional PID control algorithm is proposed by combining predictive functional control algorithm with PID algorithm.

A new performance optimized function is got by predicting a number of controlled system's future outputs. And an analytical control solution is computed by optimizing parameters in real time.

The system presented in this paper was designed so that it can work with any furnace for thermal treatments (with some adjustments in terms of power circuits realizing its interface with the technological plant).

It can practically control the furnace without being necessary to carry out setting dependent on the particular constructional features of the furnace and of the load which is subjected to the heat treatment [6].

This was actually the main goal stated in the beginning of the work paper, as there may be virtually infinite situations generated by combining different types of resistive furnaces and how the load is arranged within them. The presented will demonstrated the achievement of this goal.

II. THE ELECTRIC FURNACE WITH RESISTORS

A. The description of the plant

The plant of the resistive oven for thermal treatment taken into discussion in this paper was developed based on a personal projecting methodology.

It had been developed a furnace with a parallelepiped form, with the interior cavity of: L x B x H = 400x475x350mm. For the heating pieces, we have considered the projecting data regarding the maximum load of the oven.

There are heated simultaneously 4 steel bars having the dimensions of 210x30x40 mm. The bars are placed on a special rack, one over the other, obtaining thus 1 plate with the dimensions of 210x30x160 mm. The total power of the resistors, in this case, is 1500W. To cool the pieces from the furnace was developed a ventilation system comprising a ventilator and an exhauster [2].

In order to control this experimental platform, we have developed an embedded system, on an electronically board. This is based on a microcontroller ATMega. The connection between the furnace and the controlling board is made through the PC, whereas a Matlab-Simulink program.

In the figure below it is presented the assembly furnace – embedded system.

Fig. 1. The assembly resistive furnace – embedded system

B. The monitoring and control system

The simulations for the monitoring and automatic control of the temperature of the pieces from inside the furnace was considered for the entire one as a unit with two control inputs that provides output information about two internal sizes:

- The control input of the power supply (In 1) of the heating resistors (it was chosen a power supply with adjustable voltage in the range 0-100%);
- The control input of the ventilation system (In 2) in the range 0-100% [2].

The simulation diagram of the furnace together with the automation and control system is presented in Fig 2.

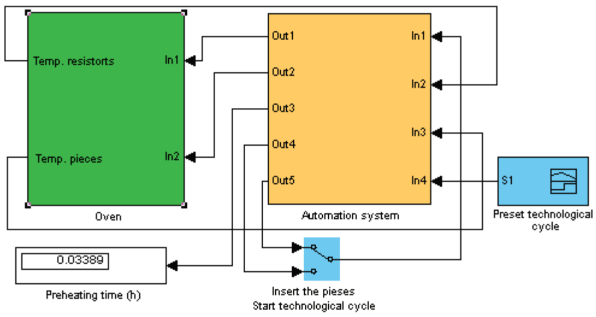


Fig.2. The diagram of the furnace and the automation system

For the control system, for the physical furnace, it has been developed a control board based on a microcontroller. The microcontroller is an Arduino Mega 2560, with 54 pins for digital input/output, 16 analog inputs, 4 serial ports, an oscillator of 16MHz, an USB port, a jack for supply and a reset button (Fig. 3.).

The control of the furnace is done through the controlling board, using the computer and Matlab/Simulink environment.

Matlab has developed a support package for the communication with Arduino. It is based on a server program uploaded in Arduino that listens the commands that came from the serial port, it executes the commands written in Matlab, and returns the results.

For the physical oven, the automation system was designed so that it can work with any furnace for heat treatment (with some adjustments in terms of circuits interfacing, achieved its technological facility).

It can actually control the furnace without being necessary to make special settings depending on the constructive characteristics of the furnace and of the load which is subjected to heat treatment.

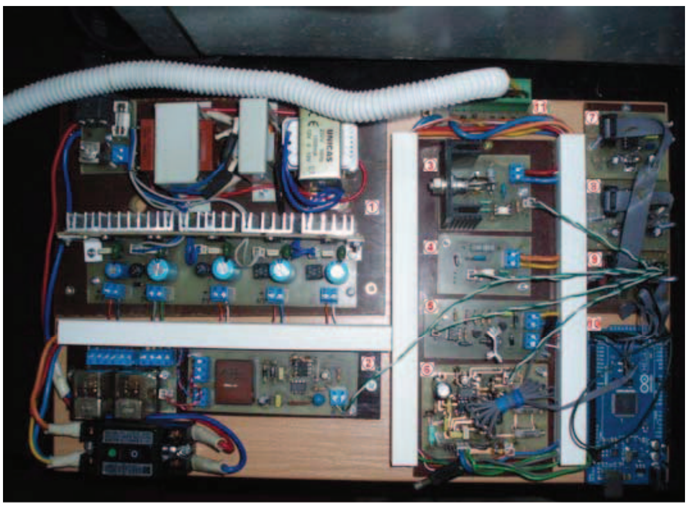


Fig. 3. The control board with microcontroller Arduino Mega 2560

We have also created a user interface, allowing us to easily modify the technological cycle, the PID coefficients for heating and for cooling, but also is very useful to monitoring the entire process: failures, start/stop of the test, etc. In this interface, created in Matlab/Simulink we are able to set the value of the preheated temperature of the furnace, the maximum temperature for the resistors and the initial temperature of the parts.

III. The Software Developped to Control the Temperature of the Parts Inside the Furnace

The use of a new plant requires the knowledge of its possibilities. First we must determine the maximum speed at which parts can be heated or cooled. The maximum heating rate is determined on one hand by the type and quantity of the parts that are subjected to heat treatment and, on the other side by the resistive elements of which the plant is equipped.

The maximum speed of cooling is dependent on the diameter of the two holes through which fresh air is introduced, and that the heated air is discharged. It also depends on the maximum flow that can be provided by the fan and blower [2].

The temperature is measured with the help of a thermocouple and a thermo-resistor that measure the temperature of the parts and the temperature of the air from inside the furnace.

After some measurements, imposing a step cycle at 250°C, we have decided to realize a technological cycle of almost 10 hours, as presented in Fig. 4.

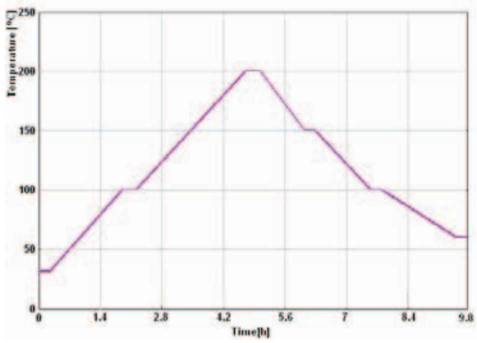


Fig. 4. Technological cycle imposed for validation

A. *User interface for monitoring and control*

Using a new installation requires knowledge of work opportunities that it has. Starting work with electro-thermal installation type must first determine the maximum speed at which parts can be heated or cooled.

These rates depend on the structural characteristics of each plant.

The maximum heating rate is determined on the one hand by the type and quantity of parts that are subjected to heat treatment and, on the other hand, by the resistive elements with which the plant is equipped.

In the construction of the experimental laboratory equipment were used three resistance-type heating elements in the tube. It is guaranteed by the manufacturer that Noted that such a resistance can operate up to a maximum of 500°C. They are typically used for heating fluids, which promotes good heat release from the surface of the resistance [5].

The maximum rate of cooling is dependent on the diameter of the two holes through which the fresh air is introduced and that the heated air is discharged. It also depends on the maximum flow that can be provided by the fan and blower [3].

In these circumstances, it was imposed a technological cycle having a maximum temperature of 250°C.

For better precision of work and for the automatic adjustment of the temperature, there are used basically all the usual methods: continuous, semi-continuous and discontinuous. Continuous adjustment methods (with P, PI, PID algorithms) are, compared with the semi-continuous and discontinuous ones, more complex and expensive, but they provide a better accuracy of the set temperature.

We have also created a user interface, allowing us to easily modify the technological cycle, the PID coefficients for heating and for cooling, but also is very useful to monitoring the entire process: failures, start/stop of the test, etc. In this interface, created in Matlab/Simulink we are able to set the value of the preheated temperature of the furnace, the maximum temperature for the resistors and the initial temperature of the parts. The interface is presented in Fig. 5.

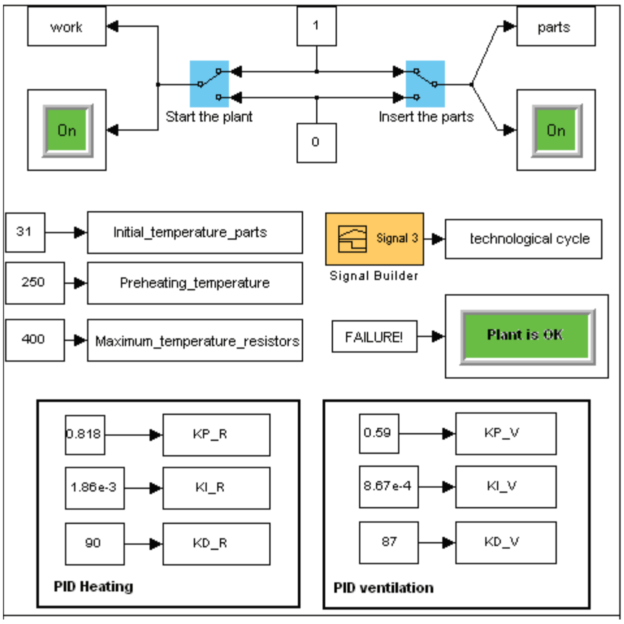


Fig. 5. Simulink model for the initialization of the plant

In order to calculate the command indices that control the power supplies of the resistors, of the ventilator and of the exhauster, we have used, for the experimental determinations, the PID regulators.

In the program developed in Matlab, in correlation with the microcontroller Arduino Mega, we have implemented the following formula for the calculation of the PID indices:

$$\Delta u(t) = K_p \left[\begin{array}{l} \left(\varepsilon(t) - \varepsilon(t-1)\right) + \frac{T_s}{T_i} \cdot \varepsilon(t) + \\ + \frac{T_d}{T_s}\left(\varepsilon(t) - 2\varepsilon(t-1) + \varepsilon(t-2)\right) \end{array} \right] \quad (1)$$

In the program, it is firstly calculated the increasing value of the control variable with this equation [3],[4]. This value is then summed with the control value given by the regulator at the previous step.

$$u(t) = \Delta u(t) + u(t-1) \qquad (2)$$

The values for the PID regulators developed for heating and for ventilation were determined with Ziegler-Nichols tuning method. In order to determine them we have used the maximal curves determined experimentally, and presented before [2].

In Fig. 6 is shown a schematic diagram of the effective communication between the temperature transducers, micro-controller and computer (Matlab programming environment), which happens during a cycle of 16s [2].

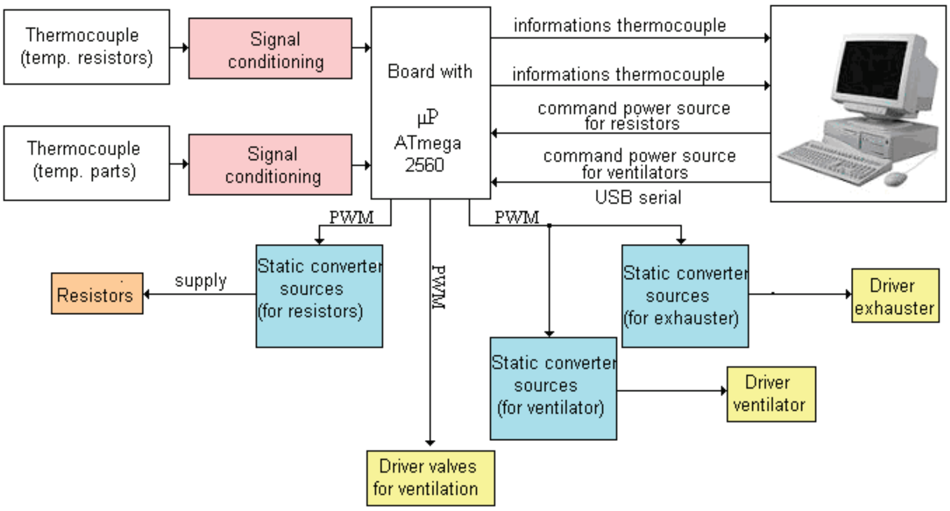


Fig. 6. The schematic diagram of the communication between the control board and the computer

Serial communication is bidirectional (Arduino sends and receives messages). It is used in general, to identifying certain programs and to interact with different peripherals [5],[8]. Most encountered functions in programming the Arduino microcontroller are void setup() {}- function that initializes the values and void loop() {} – the main loop of the program.

The most used functions of the *Serial* library are: Serial.begin(speed), Serial.read(), Serial.print() and Serial.println().

The instruction set written for the communication between the microcontroller and Matlab is presented below:

```
void setup()
{
 Serial.begin(9600);
…… }
void loop()
{ ……
Serial.println(TempTCa);
if(Serial.available()>0)
 {
  matlabdata = Serial.read();  // Read command indices
…… }
```

Asynchronous data transmission is wide frames (frames), each frame consisting of several bits. It is transmited a start bit, then a data word. It follows a parity bit, which is optional, and its role is to do a simple check of the accuracy of one or two stop bits.

In Matlab, serial port must be set (in this case using "COM6"). The command for opening the serial port is *fopen (serial)*. The command for closing the port is made with *fclose (serial)*. For the port must be set a number of parameters such as:

- *Baud rate* must be the same as the one set in the Arduino (9600);
- *Terminator character* (which marks the end of communication). In this case there was used "LF".
- *Timeout* – Specifies the maximum wait time allocated for performing a read or write operation; In this case we used 60s.
- *InputBufferSize* – it sums up the total number of bytes that can be stored in the input buffer;
- *OutputBufferSize* - it sums up the total number of bytes that can be stored in the output buffer;
- *ReadAsyncMode* - to set the transmission mode: synchronous or asynchronous. In this case we chose for an asynchronous transmission on "manual";

Practical instruction sequence used to set parameters for the serial communication is:

```
arduino=serial('COM6');
arduino.BaudRate=9600;
arduino.InputBufferSize=18;
arduino.OutputBufferSize=1;
set(arduino,'terminator','LF','Timeout',60);
arduino.ReadAsyncMode='manual';
fopen(arduino);
```

Matlab transmits data to microcontroller with the command [7],[8]:

fwrite (Arduino coefTR, 'uint8', 'async'), which transmits data in binary format.

To read data from the microcontroller we used the command:

data = fscanf (arduino, '%f');

The logical diagram that shows the functionality of the microcontroller is presented in ANNEX. The main program is written and uploaded in the microcontroller Arduino Mega 2560, an it works in tandem with the program written in Matlab.

Practically, the program from Matlab calculates the coefficients for heating and ventilation, and the program written in the microcontroller sends different commands to the actuators (valves, resistors, ventilator, exhauster, etc.)

IV. RESULTS AND CONCLUSIONS

The system proposed in this paper provides a good tracking of the prescribed temperature from the technological cycle.

In Fig.7. it can be observed that momentary differences between the two temperatures are below 1,75°C throughout the entire range of the test. These values are very good given the literature requirements that specify for allowable differences in heat treatment of metal between ± 5÷10°C.

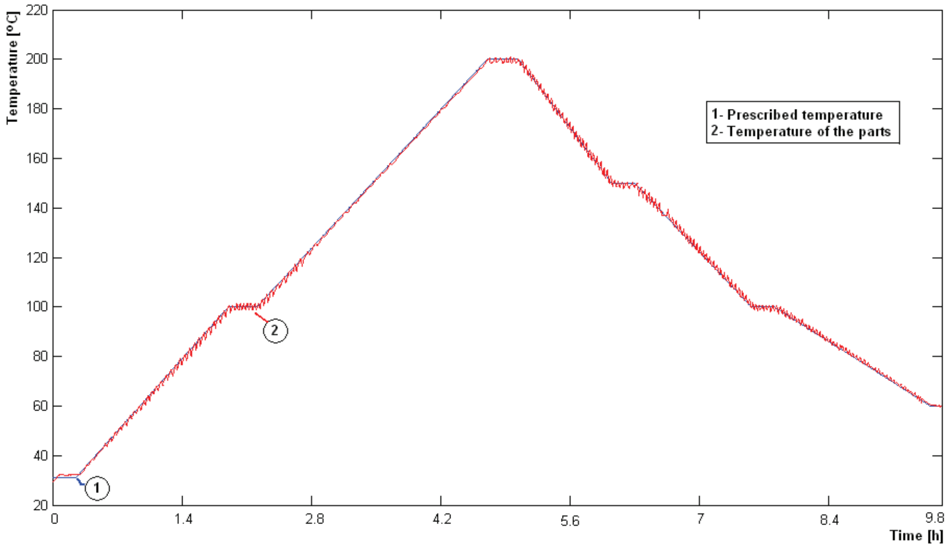


Fig. 7. Prescribed temperature and the temperature of the parts measured at experimental furnace

This paper is part of the current trends in development of systems for monitoring and control of industrial indoor ambient conditions.

The field of heat treatments is a timeless zone within the industrial processes, especially since recently it is the trend to upgrade materials used in the construction of assemblies and subassemblies and discovery of materials with superior properties.

There have been several attempts in order to provide optimal coefficients of the two controllers.

We always had in mind that we have to insert in the experimental plant a computer system type PC with the following features:

- to allow easy insertion of required technological cycle;

- to allow convenient modification of control parameters;

- to ensure the calculation of the control coefficients;

- to display in real-time the evolution of the temperature of the parts and of the resistance depending on the temperature imposed by technological cycle;

- to store the results in order to monitor the installation and make subsequent analyzes.

REFERENCES

[1] Miguel A. Zamora-Izquierdo, José Santa and Antonio F. Gómez-Skarmeta, "Integral and Networked Home Automation Solution towards Indoor Ambient Intelligence" , Pervasive Computing, IEEE (Volume:9 , Issue: 4), ISSN: 1536-1268.

[2] NEACĂ, A.M., NEACĂ,M.I., Methods of Automatic Control applied to a Resistive Furnace for Thermal Treatments, 9th International Conference on Electromechanical and Power Systems – SIELMEN 2013, October 15-16, 2013, Iaşi, România, October 17-18, 2013, Chişinău, Moldova, CD_Proceedings, ISBN 978-606-13-1560-4, pg. 147-151.

[3] A.M. Neacă, Comparison between Different Techniques of Controlling the Temperature inside a Resistive Oven, IEEE 16th International Conference on System Theory, Control and Computing, Sinaia, Romania, October 12-14, 2012.

[4] Mhaskar P., El-Farra N. and Christofides P., A Method for PID Controller Tuning, Using Nonlinear Control Techniques, Proceeding of the 2004 American Control Conference, Boston, Massachusetts, 2004.

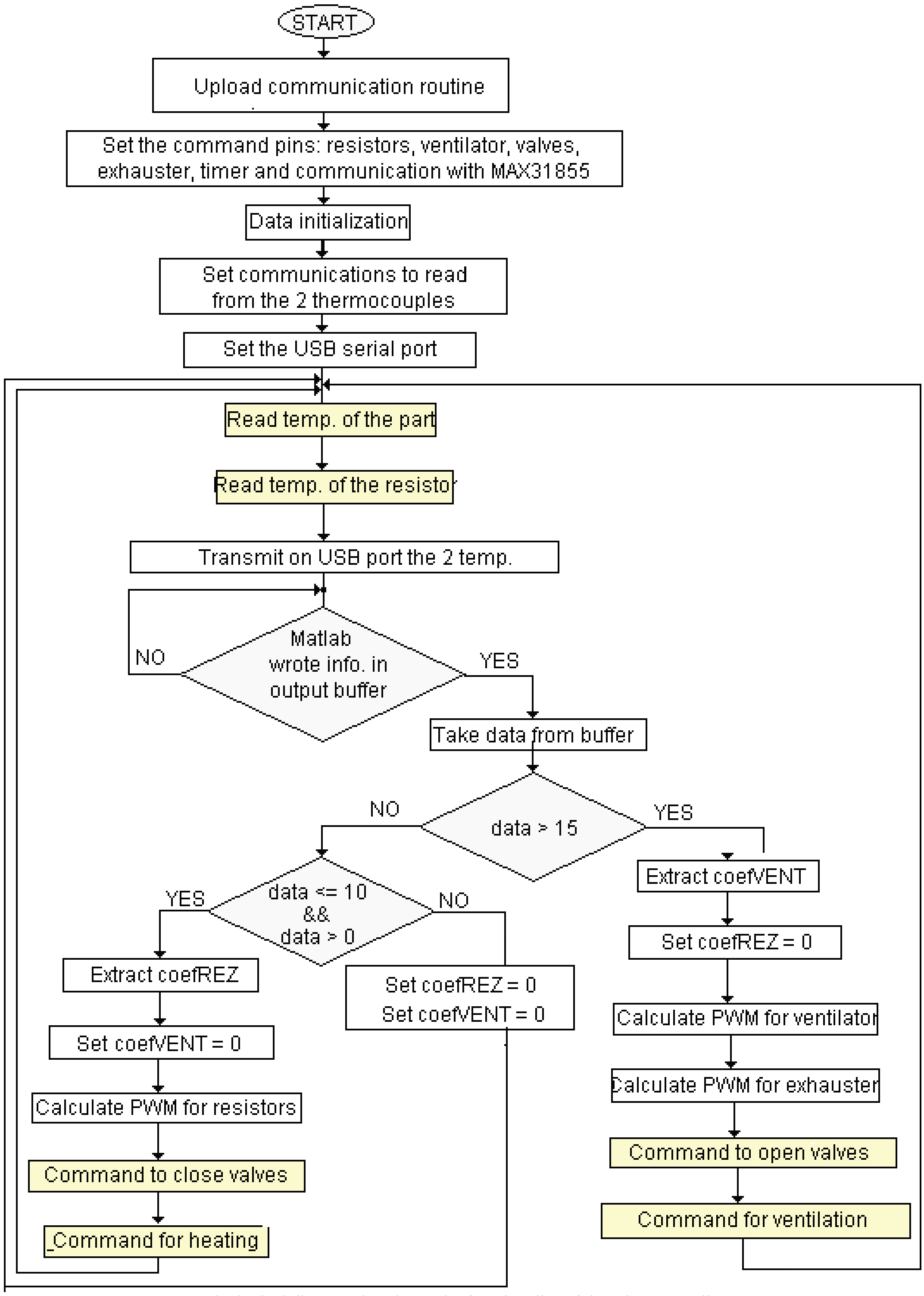
[5] NEACA, M.I, NEACĂ, A.M., High Temperature Measurements in Industrial Equipment, Analele Universității din Craiova, Seria Inginerie electrică, Anul 36, nr.36, 2012, ISSN 1842-4805, pag.278-283.

[6] Banerjee S., Sanyal D., Sen S., Puri .K., A methodology to control direct-fired furnaces, International Journal of Heat and Mass Transfer 47, 5247–5256, 2004.

[7] Emine Dogru Bolat, Implementation of Matlab-SIMULINK Based Real Time Temperature Control for Set Point Changes, International Journal of Circuits, Systems and Signal Processing, Vol. 1, 2007.

[8] Mangal A., Serial Communication between Arduino and Matlab, IIT Bombay, June 2012.

ANNEX



The logical diagram that shows the functionality of the microcontroller