Implementation of a Fuzzy Logic-Based Embedded System for Temperature Control

Gabriel Oltean[1)] and Laura-Nicoleta Ivanciu[1)]

[1)] Bases of Electronics Department, Technical University of Cluj-Napoca, Romania
Laura.Ivanciu@bel.utcluj.ro

Abstract: *The core of this work consists in an explicit and complete implementation of an embedded system for temperature control. The embedded system is developed around an Arduino board and uses a fuzzy logic system as controller. The literature abounds in various solutions referring to design, implementation, and/or simulations of control systems, fuzzy logic controllers, embedded systems, and their combinations. This implementation offers to novice designers a platform for understanding, experimenting and learning the main concepts of both embedded system and closed loop control system, operating in a real environment. The major achievements of this work can be summarized as follows: access for measurements in relevant points of the system, illustration of the use of some fundamental devices and concepts in modern electronics (smart sensor for temperature measurement, power control using SCR, DAC converter, C++ programming, implementation of a fuzzy logic controller, I^2C and 1-Wire interfaces), possibility to create different operating scenarios (by changing the set point temperature, sampling period, and scaling factors). The experimental results, in various scenarios, substantiate the expected operation of our temperature control system.*

1. INTRODUCTION

A system whose principal function is not computational but is controlled by a computational system embedded within it is referred to as an embedded system [1].

In the field of control, fuzzy logic has become very popular, mainly because the process of fuzzy logic control is simply to put the realization of human control strategy, where conventional control heavily relays on appropriate mathematical model.

There are a lot of approaches referring to the utilization of fuzzy logic control in the frame of embedded systems. An adaptive fuzzy controller in which the scaling factors of the membership functions are adapted in real time using a reinforcement Q-learning algorithm is implemented on an Arduino DUE board to control a DC motor with flexible shaft [2]. The development and implementation on an Arduino Due board of a fuzzy logic controller operating in real time, for speed control of a dc motor, is presented in [3]. Automated temperature control systems based on microcontrollers are discussed in [4], [5], [6]. To achieve real-time operation in digital video processing, an algorithm for video de-interlacing using three fuzzy-logic systems is implemented as a hardware IP core on a FPGA-based embedded system [7]. In [8], a non-linear liquid level process is controlled through the medium of a Sugeno fuzzy logic controller running on an Arduino Mega board.

Another common approach is to run the fuzzy logic controller in the Matlab/Simulink development environment due to the existence of the dedicated Fuzzy Logic Toolbox. For example, in [9] a liquid level control system is developed, where there is a real process, a microcontroller responsible for the sensors read, data transmission and PWM control of actuators while the fuzzy logic controller runs in Matlab. Besides, a lot of studies are entirely (both process and control) based on Matlab/Simulink simulation: fuzzy logic control of a dc motor speed [10] or a fuzzy control algorithm to guarantee high-efficiency output of photovoltaic batteries.

In this paper, we propose a solution that integrates the main concepts referring to embedded systems, closed-loop control, fuzzy logic system implementation, smart sensors, communication interfaces, and actuators in an easy-to-understand and self-explanatory practical implementations.

The purpose of this implementation is to support novice designers in the learning process, by providing them with a platform for understanding and experimenting the main concepts of both embedded system and closed loop control system, operating in a real environment. The proposed embedded system allows access for measurements in relevant points of the system, illustration of the use of some fundamental devices and concepts in modern electronics (smart sensor for temperature measurement, power control using SCR, DAC converter, C++ programming, implementation of a fuzzy logic controller, I^2C and 1-Wire interfaces), and the possibility to create various operating scenarios (by changing the set point temperature, sampling period, and scaling factors).

2. Overview of the Proposed System

The block diagram of the system is presented in Fig. 1. The process consists in maintaining a user specified constant temperature (T_{ref}) in a thermal enclosure. The current temperature, T is measured using a smart sensor DS18B20 which provides a 9 to 12-bit temperature reading over a 1-Wire interface [11].

The modification of the temperature is controlled by means of a heating resistor whose AC heating power is modulated by a phase-control SCR. The SCR is driven by a specialized TCA785 [12] circuit located on the phase control board. The analog control voltage required by TCA785 is provided by the 12-bit digital-to-analog converter MCP4725 [13] that communicates with the digital part of the control system on a two-wire I^2C™ compatible serial interface. The necessary voltage adaptation between the DAC and the phase control circuit is assured through a voltage amplifier based on a rail-to-rail AD820 operational amplifier.

The Arduino Uno board [14] is the "brain" of the entire system. It is primarily responsible for the update of the digital control signal u, at every time instance. Therefore, the actual temperature, T_k is read and the actual temperature error (err_k) and change of temperature error ($cerr_k$) are updated, as follows:

$$\begin{aligned} err_k &= T_k - T_{ref} \\ cerr_k &= err_k - err_{k-1} \end{aligned} \tag{1}$$

where err_{k-1} is the temperature error in the previous time instance.

The star of the entire system is the fuzzy logic controller, whose role is to infer the best modification in the control signal, in every time instance Δu_k. The digital version of the actual control signal u_k is

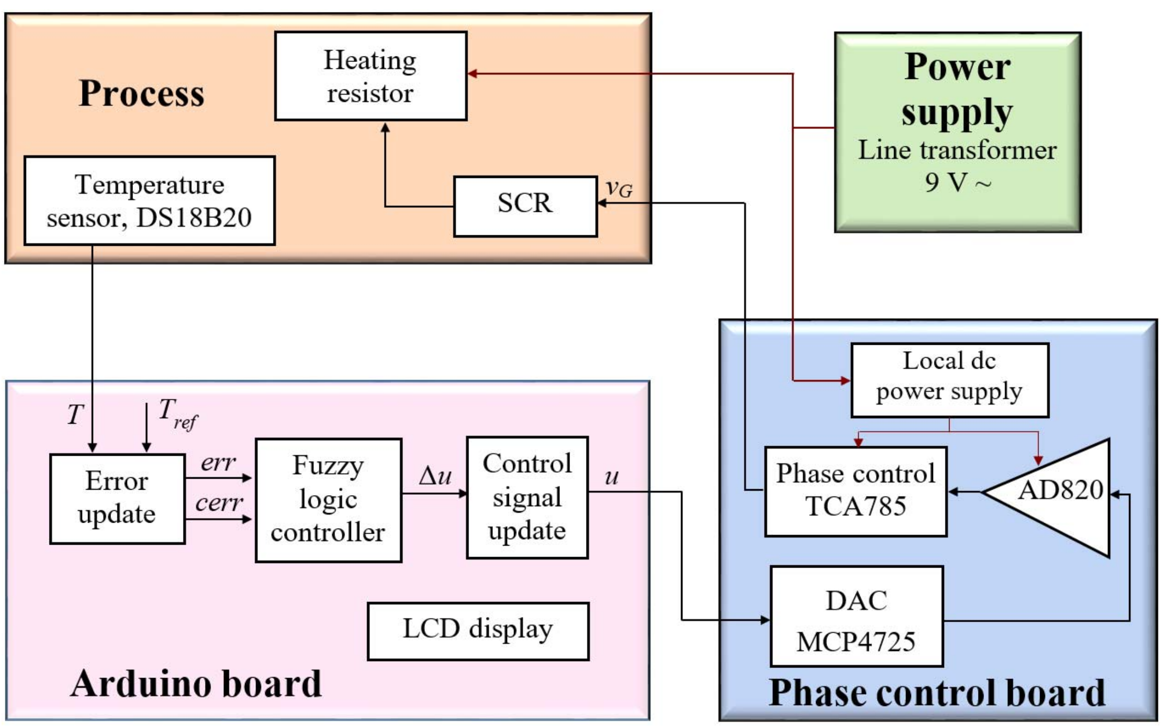

Fig. 1. Block diagram of the fuzzy logic-based embedded system for temperature control.

updated using the relation:

$$u_k = u_{k-1} - \Delta u_k \qquad (2)$$

The power supply comes from a step-down line transformer that provides 9 V_{rms} AC voltage in its secondary winding.

3. System Implementation

Our proposed system contains two main parts: the power circuit and the control circuit. The power circuit is presented in Fig. 2. *R* is a 2 Ω heating resistor connected to the secondary windings of a line transformer that provides 9 Vrms AC voltage.

The mean power dissipated by the resistor is controlled by the *Q* SCR according with its on/off state. Thus, if the *Q* is triggered near the start of each positive half-wave (around 0^o phase angle) the mean power dissipated by *R* is near the maximum possible, while if the *Q* is triggered near the end of each positive half-wave (around 180^o phase angle), the dissipated power tends to be zero. The SCR is triggered by the positive pulse of the voltage applied in its gate, v_G, provided by the control circuit.

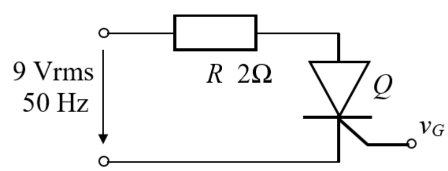

Fig. 2. The power circuit.

3.1. The Control Circuit

The control circuit is presented in Fig. 3, and it is developed around the fuzzy logic controller, that has two inputs (*errFls* and *cerrFls*) and one output (Δ*uFls*). The range of value for all these three variables is the same, [-1; 1].

The current temperature error (*err*) and change in temperature error (*cerr*) are determined according with equation (1). For the flexibility of the control system, two scaling factors S_e (for *err*) and S_c (for *cerr*) were introduced. By changing the scaling factor, the user can easily modify the behaviour of the control circuit, the fuzzy logic controller being more or less sensitive to each input; the larger the scaling factor, the bigger the importance of the corresponding variable. To impose upper (+1) and lower (-1) bounds on the input signals, a saturation block is used for each input, so that it clips the signal, when its values surpass the [-1; 1] range.

The fuzzy logic controller generates the Δ*uFls* signal that is further multiplied with the scaling factor *Su* obtaining the final modification of the control signal, Δ*u* at each time instance. The actual value of the control signal, *u* is than computed according with equation (2). The range of variation for the digital control signal is limited between 0 and 4095 by the saturation block.

The DAC circuit (12 bits) converts the digital control signal to an analog signal, which multiplied by the voltage amplifier, finally generates the control voltage u_a for the phase control specialised circuit.

The phase control circuit (around the TCA785 IC), according with its control voltage u_a, generates the positive voltage pulses v_G that trigger the SCR. It is worth to mention the qualitative relation introduced by the control circuit. For example, if the temperature should increase, Δ*u* increases, *u* decreases, u_a decreases, v_G decreases, and finally the mean power dissipated by the heating resistor increases as well, leading to the expected temperature increase.

3.2. The Fuzzy Logic Controller

The fuzzy logic controller was implemented on the Arduino board in the open-source Arduino Integrated Development Environment (IDE) [14]. The system is a first-order Takagi-Sugeno one, with two inputs *errFls* and *cerrFls* and one output Δ*uFls*. For all three variables, the universe of discourse is [-1; 1]. For the inputs, the fuzzy sets are triangular (Fig. 4), while for the output the fuzzy sets are singleton (Fig. 5.)

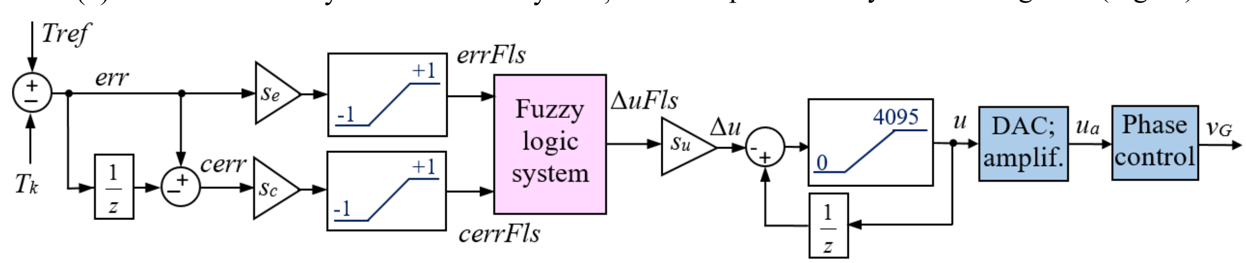

Fig. 3. The control circuit.

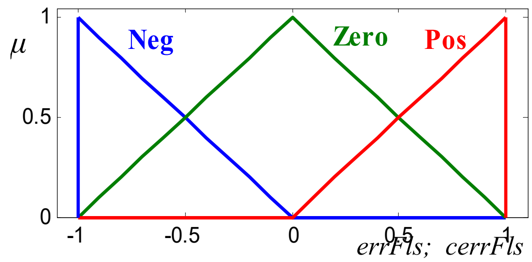

Fig. 4. Fuzzy sets for the inputs.

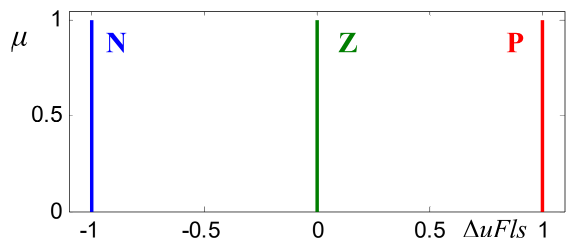

Fig. 5. Fuzzy sets for the output.

The rule base contains 9 fuzzy rules, represented in Table 1. The defuzzification method, used to transform the partial output fuzzy sets resulted from the inference process into a crisp value is the weighted average method. The control surface that illustrates the full operation of the fuzzy controller is presented in Fig. 6.

It is worth to mention that despite its simplicity (only 9 rules) the fuzzy system performs very well, providing the right output for any combination of input values. In this approach, we choose to use a normalised fuzzy system. To change the behaviour of the controller as a whole, no intervention should be made on the fuzzy sets, but rather on the three scaling factors, as explained in section 3.1.

Table 1. Rule base of the fuzzy logic system

		errFls		
		Neg	Zero	Pos
cerrFls	Neg	N	N	Z
	Zero	N	Z	P
	Pos	Z	P	P

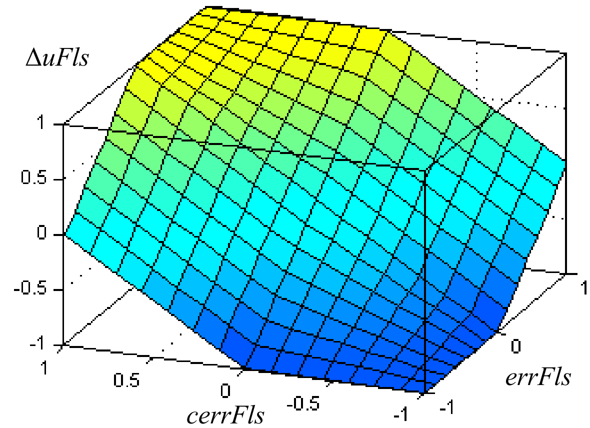

Fig. 6. Control surface of the fuzzy logic controller

4. Experimental Results

The operation of the entire system was verified on a real-time setup for different scenarios by the step response of the system for a user-specified set point temperature. Fig. 7 presents the system response for a set point temperature $T_{ref} = 37^{\circ}\mathrm{C}$ for an environmental temperature of $27.4^{\circ}\mathrm{C}$. The scaling factors in this case are: $S_e = 0.15$, $S_c = 1.5$, and $S_u = 1000$. As expected, the system response is a typical one, presenting some decreasing positive and negative overshoot. The main parameters characterizing the time response are:

- rise time = 216.5 s;
- settling time = 550 s;
- positive overshoot = 0.5ºC,
- negative overshoot = 0.25ºC;
- steady state error = 0.06ºC.

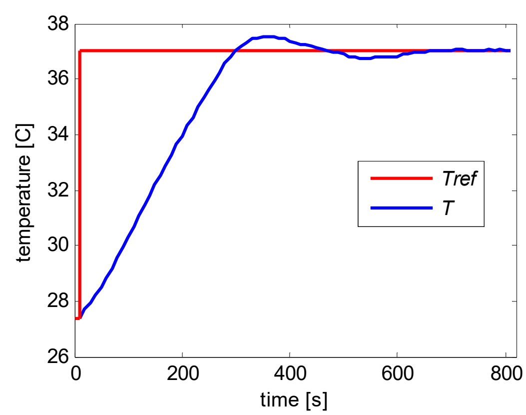

Fig. 7. Control surface of the fuzzy logic controller

All intermediate variables in the control system can be plotted to be further analyzed for a deep understanding of the control circuit (Fig. 8). For example it can be seen that the modifications of the control signal Δu are large positive in the beginning (1000 and close to 1000), when both *err* and *cerr* are large positive. As the system advances toward the steady state (after about 500 s) Δu have small positive and negative magnitude. The digital control signal, u varies between 0 (meaning full mean heating power) and 4000 (4096 meaning no heating power at all). The range of variation for the analog control signal, u_a is [0; 10][V]. The upper bound is set by the rail-to-rail DAC, with 3.3 V as the full-scale output voltage (supplied at 3.3 V) and the gain of the analog amplifier (around 3). In the steady-state regime the control signals are $u = 3167$ and $u_a = 7.65$ V .

The behaviour of the phase control circuit can be illustrated by some waveforms using an oscilloscope. Fig. 9 contains the following waveform (a four-channel oscilloscope):

- the analog control voltage applied at pin 11 of the TCA785 IC, 4.8 V (ch 3, magenta);
- the ramp voltage, generated by the TCA785 IC, at pin 10 (ch2, light blue);
- the positive voltage pulse generated by the TCA785 at pin 15, to be applied in the gate of the SCR to set it on (ch1, yellow); the voltage pulse is generated when the ramp voltage exceeds the analog control voltage;
- the almost sinusoidal supply voltage, in the secondary of the line transformer (ch4, green); the moment when the SCR switches on (when the positive pulse appears in its gate) is obvious on the waveform – the voltage decreases due to the large current ensured through the 2 Ω heating resistor (4.5 A peak value);

The behaviour of the power circuit (see Fig. 2) is illustrated by the waveforms in Fig. 10, using two input channels and a math channel:

- the supply voltage, in the secondary of the line transformer (ch1, yellow);
- the voltage drop across the SCR (ch2, blue);
- the voltage drop across the heating resistor (math ch, red);

For the case presented above, the phase angle is 95 °, corresponding to a mean heating power around half of the maximum possible. The maximum mean power is obtained when the phase angle is close to zero (the analog control voltage close to 0 V), while zero heating power appears if the phase angle is close to 180 ° (the analog control voltage close to 10 V).

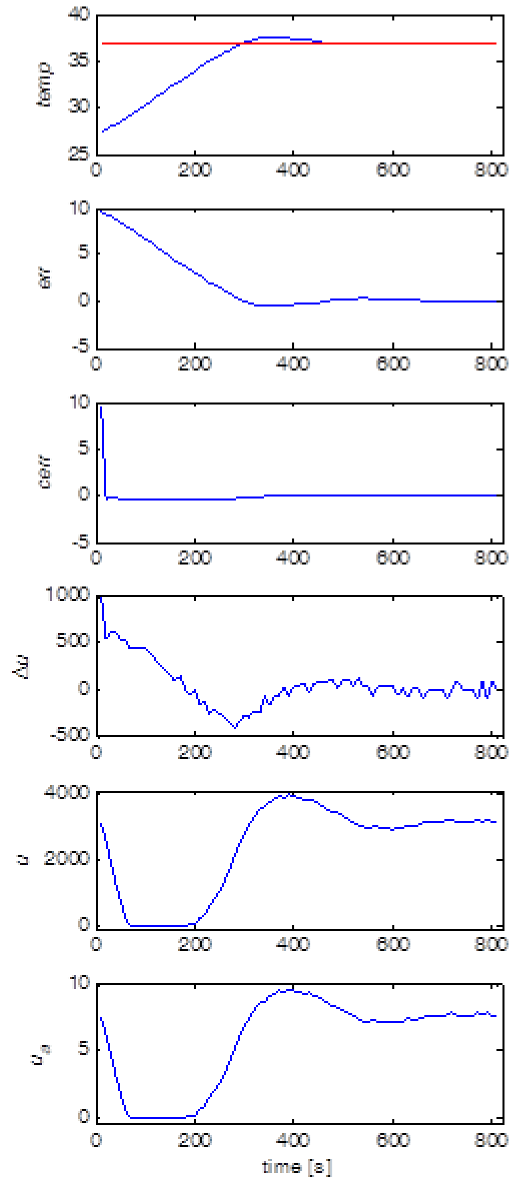

Fig. 8. Time evolution of some intermediate variables illustrating the control system operation.

The setup is very flexible from the point of view of user access and possibility to set different operating scenarios by simply modifying some constants in the software code and starting again the control process. The constants that can be modified, one at a time or in any combination are: scaling factor for temperature error; scaling factor for change in temperature error; scaling factor for the modification of control signal; the reference temperature; the sample period used for reading a new temperature value and re-computing a new control voltage. Therefore, the user can experiment different configuration of the control system to understand the effect of each parameter, and to optimize the time response, even towards an overdamped or undampened response.

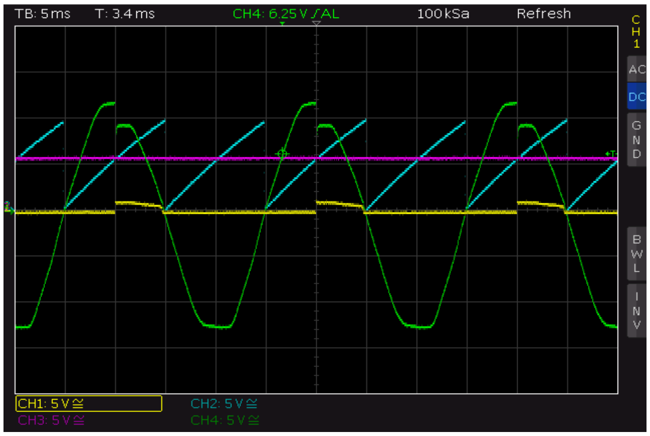

Fig. 9. Waveforms for the phase control circuit.

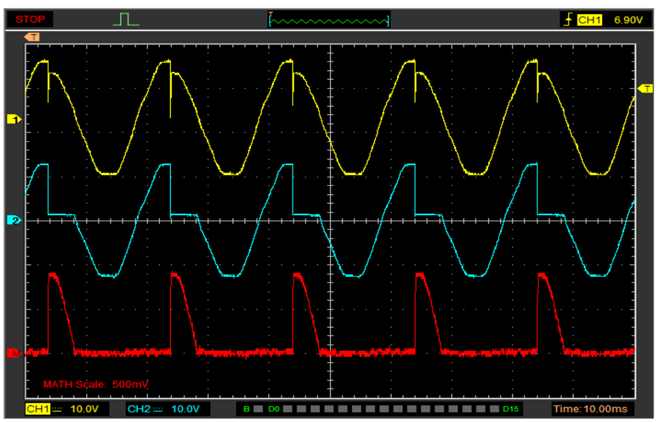

Fig. 10. Waveforms for the power circuit.

5. CONCLUSIONS

Understanding theoretical concepts is highly facilitated by experimenting the studied phenomena or process in real environments, and embedded systems and closed loop control systems make no exception to this rule. This paper presents the explicit and complete implementation of an embedded, closed loop control system, for temperature control. The embedded system is developed around an Arduino board and uses a fuzzy logic system as controller. The implementation makes use of various modern electronic concepts (smart sensor for temperature measurement, power control using SCR, DAC converter, C++ programming, implementation of a fuzzy logic controller, I2C and 1-Wire interfaces), while maintaining a high level of flexibility. By making minor changes in the software code, Different operating scenarios can easily be employed, so that the user acquires a thorough understanding of the effect of each parameter and obtains an optimized time response.

REFERENCES

[1] A. M. Ibrahim, “Fuzzy Logic for Embedded System Applications”, Elsevier, 2004.

[2] A. A. Khater, M. El-Bardini, N. M. El-Rabaie, "Embedded Adaptive Fuzzy Controller Based on Reinforcement Learning for DC Motor with Flexible Shaft", *Arab J. Sci. Eng*., vol. 40, no. 8, 2015, pp. 2389–2406.

[3] H. R. Jayetileke, W. R. de Mei, H. U. W. Ratnayake, "Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due," *7th International Conference on Information and Automation for Sustainability*, Colombo, 2014, pp. 1-6.

[4] A. Pimpalgaonkar, M. Jha, N. Shukla, K. Asthana, “A Precision Temperature Controller using Embedded System”, *Intl. J. of Sci. and Res. Pub*., vol. 3, no. 12, 2013, pp. 1-4.

[5] S.R.Sharma, P.B.Dahikar, "Embedded Design of Temperature Controller using PIC16F876A for Industries and Laboratories", *Intl. J. of Innov. Res. in Comp. and Comm.Eng.,* vol. 1, no. 10, 2013, pp. 2414–2422.

[6] P. Singhala, D.N. Shah, B. Patel, “Temperature Control using Fuzzy Logic”, *International Journal of Instrumentation and Control Systems,* vol.4, no.1, January 2014, pp. 1- 10.

[7] P. Brox, I.Baturone, S.S.-Solano, "Fuzzy logic-based embedded system for video de-interlacing", *Appl. Soft Computing,* vol. 14, Part. C, pp. 338–346, 2014.

[8] B.Çelik, S.Birtane, E.Dikbıyık, H.Erdal, "Liquid Level Process Control with Fuzzy Logic Based Embedded System", IEEE 9th Intl. Conf. on Electrical and Electronics Eng. pp. 874–878, 2015.

[9] J. G. Guarnizo, J. Avendano, "Liquid level system as a pedagogical tool to teach fuzzy control", *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Cholula, Mexico, 2017, pp. 1-5.

[10] Y. A. Almatheel, A. Abdelrahman, "Speed control of DC motor using Fuzzy Logic Controller," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE),* Khartoum, 2017, pp. 1-8.

[11] DS18B20 Datasheet, https://goo.gl/ibYqo4

[12] TCA785 Datasheet, https://goo.gl/cF7IdJ

[13] MCP4725 Datasheet, https://goo.gl/487Uen

[14] Arduino website, https://www.arduino.cc/