IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

A Multiple Temperature-Control And Alarm System, Based On LM35 Sensor, Using FPGAs and VHDL

Dr Evangelos I. Dimitriadis[1], Leonidas Dimitriadis[2]

Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, End of Magnisias Str, 62124 Serres Greece[1]

Undergraduate Student, Department of Information and Electronic Engineering, International Hellenic University, 57400, Sindos Thessaloniki, Greece[2]

Abstract: A multiple temperature-control system, based on LM35 sensor, FPGAs and VHDL, is presented here. The system is capable of providing a series of controls and subsequently activate respective alarm systems. It can simultaneously monitor three basic temperature-related parameters. The first is temperature range values of specific area or a human. Blue, green and red LEDs light up, to present temperature lowering below lower limit value, temperature remaining within set values or exceeding upper set value, respectively. If temperature is out of limits buzzer also sounds. Second basic parameter monitored here, is temperature rising or lowering rate within specific time set values and subsequent activation of center yellow LED and half right of board LEDs or center white LED and half left of board LEDs, respectively. Finally the third basic parameter monitored with our system, is temperature remaining above upper critical set value or below lower critical set value for a specific time period, thus activating breadboard's right yellow or white LEDs, respectively. LM35 temperature sensor used here and its analog voltage values act as input to FPGA's ADC unit and converted temperature values are presented to seven-segment displays. All LED systems activated here, correspond to related external control systems which are activated on a case-by-case basis. The system uses DE10-Lite FPGA board and taking into account that specific time periods, as well as upper and lower temperature limits and temperature rising or lowering rates can be set to a variety of values, gives our system the ability of implementation in a wide range of applications such as patient, room, laboratory, industrial or external environment monitoring.

Keywords: LM35 sensor, Temperature, FPGA, VHDL, Buzzer, LEDs.

I. INTRODUCTION

FPGAs provide the main advantage of combining software and hardware, thus having the ability of hardware programming for a series of applications. Languages used for FPGAs' programing are VHDL and Verilog and VHDL is the one used in our work.

An interesting application field of FPGAs is temperature monitoring and controlling, incorporating also various alarm systems. [1-5] Presented works deal with topics such as monitoring temperature in FPGA based SoCs, FPGA based real-time remote temperature measurement system, FPGA based temperature control and monitoring system for X-ray measurement instrument, FPGA Alarm System Based on Multi Temperature Sensor and Temperature Sensors in FPGA Based On Digital Nonlinear Oscillators for Improved Sensitivity. All the above works use complicated systems and some of them also expensive, but the problem of presenting a simple system, capable of providing monitoring, measuring and controlling basic temperature-related parameters, is not clearly solved. Our system uses the cheap and reliable temperature sensor LM35, thus providing a simple solution for monitoring and controlling various temperature parameters in conjunction with the activation of specific alarm systems. This work presents a system dealing with an important triple of temperature measurements and controls. Temperature range values, temperature rise and lowering rates within specific time intervals and temperature overcoming set values or lowering below set values for a number of times, within specific time periods. Another benefit of our system is that it can work with a variety of temperature sensors which provide an analog output and also its cost is remarkably low.

II. DESIGN OVERVIEW AND OPERATION OF THE SYSTEM

Figure1 presents device overview and operational units of our system, using FPGA DE10-Lite board, while Figure 2 presents circuit diagram of the system.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

It is obvious from both of the above figures that our system, except from DE10-Lite FPGA board, contains also some basic circuit parts. We can see three LED systems. First is temperature monitoring system, second is temperature increase or decrease rate monitoring unit and third is temperature above or below critical values counting-monitoring system. We can also observe buzzer alarm system and LM35 temperature sensor unit. DE10-Lite FPGA board used here offers, its seven-segment displays for presenting input LM35 sensor voltage values converted to input temperature in Celsius degrees and board LEDs mentioned above. Figure 3 presents all the above units.
Time is the other input value used here and it is provided by FPGA's clock.

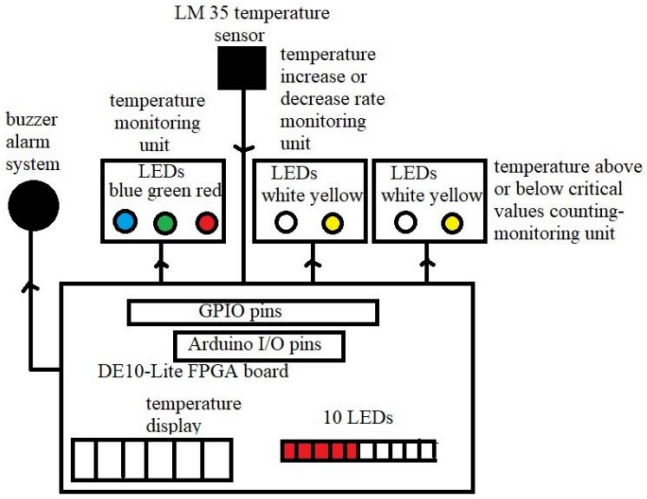

Figure 1: Device overview and operational units of our system.

Our multiple temperature-control system starts operating as soon as power supply +5V is applied to all circuits and the VHDL program is sent via USB Blaster interface, to FPGA chip. Input values from both LM35 unit and FPGA's clock are entered in our system. Analog to digital converter (ADC) of DE10-Lite proceeds to conversion and finally input voltage values are given their calculated corresponding Celsius degree values, which are finally presented in seven-segment displays of the FPGA board.

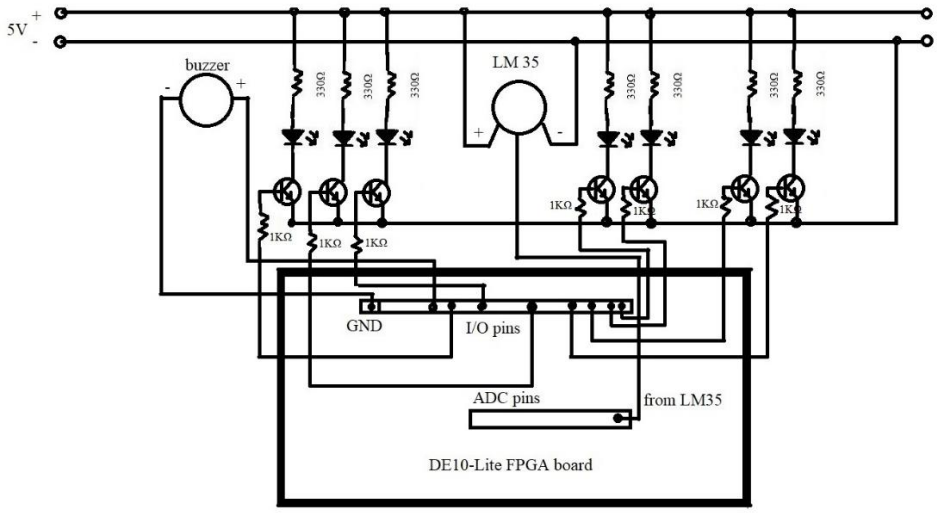

Figure 2: Circuit diagram of our system

Simultaneously three basic processes start running. At first, temperature monitoring and controlling unit receives LM35 sensor's input values and checks whether temperature values are within or out of set limits. We chose a lower temperature limit value of 18ºC and an upper limit of 30ºC. If input temperature values are below lower limit then blue external LED lights up and buzzer starts sounding. In case that temperature value is within set limits then green external LED lights up and buzzer stops sounding. Finally if temperature value is above upper set limit then red external LED lights up and buzzer starts sounding again. The above external LEDs are connected to I/O pins of the FPGA board and act as outputs for our system.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

The second unit that is activated upon system's operation is temperature increase or decrease rate monitoring unit. This unit uses both LM35 and time input values. Its role is to monitor the possibility of a rapid temperature rising or lowering rate. If increase temperature rate overcomes set value then yellow LED of the unit lights up with simultaneous lighting of half right board LEDs. In case that decrease temperature rate

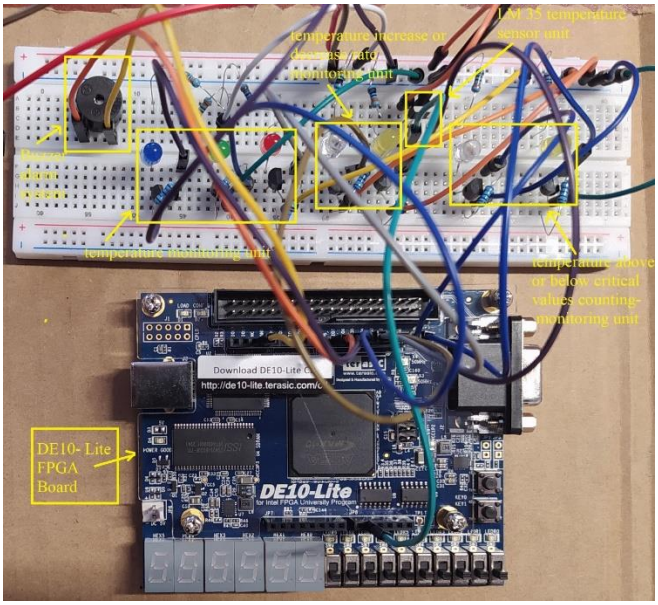

Figure 3: The multiple temperature-control system of this work.

overcomes set value then white LED lights up with simultaneous lighting of half left board LEDs. It must be mentioned that specific time periods and increase or decrease temperature rates are set by the programmer. This gives our system the ability to monitor a human or an environment for a variety of time periods. In this work in order to obtain fast results we observe increase or decrease temperature rates from first 30sec to 60sec time with a temperature value difference of 3°C. We must mention that all LED systems, as well as buzzer unit of this work, play the role of outputs which could activate corresponding control systems.

Another important fact of our system's design is that it receives input voltage values periodically, ensuring continuous temperature change monitoring.

The third unit that is activated upon system's operation is the one that observes

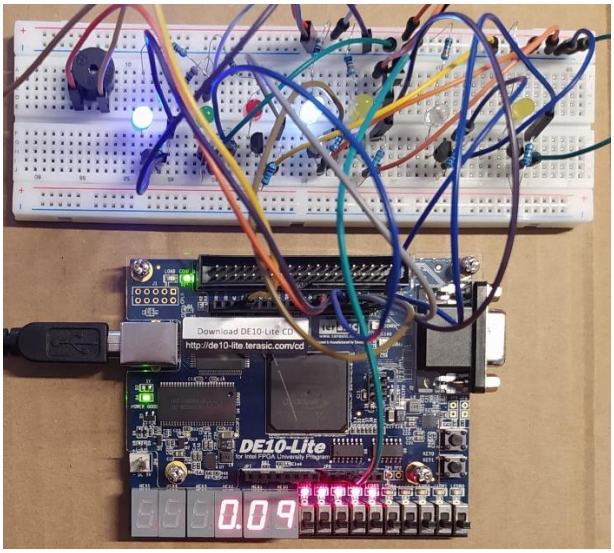

Figure 4a: The multiple temperature-control system of this work, operating. Temperature is below lower set limit and temperature rate decrease exceeds set value.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

whether the monitored environment remained for a specific time period from 120sec to 240sec, within temperature values below lower set value or above upper set value.

Essentially we are using a counter unit which counts with clock pulses, the times that input temperature values are found to be above upper set limit or below lower set limit. In the first case, at the right of the circuit, yellow LED lights up, while in second case white LED lights up. We must mention that, for simplicity reasons, we set the number of counted times exceeding upper or lower set temperature values, equal to 3, meaning that three clock pulses with simultaneous exceeding temperature values are enough to cause white or yellow LEDs of this unit to light up.

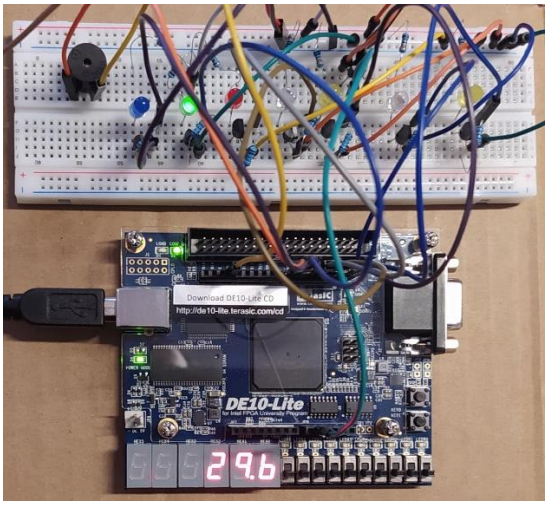

Figure 4b: The multiple temperature-control system of this work, operating. Temperature is within programmer set upper and lower values.

All external LEDs use an NPN 2N2222 transistor as a switch, with its base electrode connected to FPGA's I/O pins which are our system's outputs.

Figures 4a, 4b, 4c and 5 present different phases of our system's operations, mentioned above. Figure 4a shows that temperature is below lower set limit and temperature rate

decrease exceeds set value. This results to blue LED lighting at the left of the circuit showing that temperature value is below lower set limit. It can also be seen that due to temperature rate decrease exceeding set value, both left half of board LEDs and center white LED light up, as we mentioned above concerning our system's operation.

Figure 4b presents temperature being between programmer set upper (30ºC) and lower (18ºC) values and consequently green LED lights up.

Figure 4c shows that measured temperature exceeds upper set value, so red LED lights up.

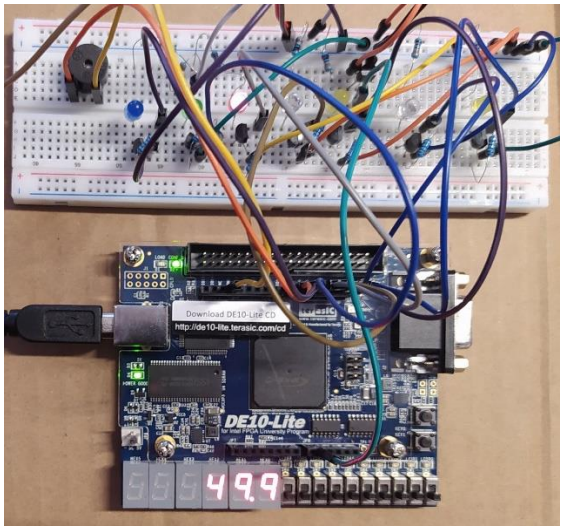

Figure 4c: The multiple temperature-control system of this work, operating. Measured temperature exceeds upper set value.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

In Figure 5 we observe that all three monitoring units of our system, are in operation mode and they have been affected by temperature changes. At first, temperature value is above upper set limit causing red LED lighting. Second observation shows that temperature decrease rate exceeds set value, leading to both center white LED and half left of board LEDs lighting. Initial temperature value of 49.9ºC shown in Figure 4c, reduced to a value of 31.5ºC presented in Figure 5, but decrease rate exceeded the value of 3ºC in a time period of 30sec, set in VHDL program. Third observation makes obvious that temperature values remained higher than upper set limit, for specific time set period extending from 120sec to 240sec, leading to yellow LED lighting at the right of breadboard.

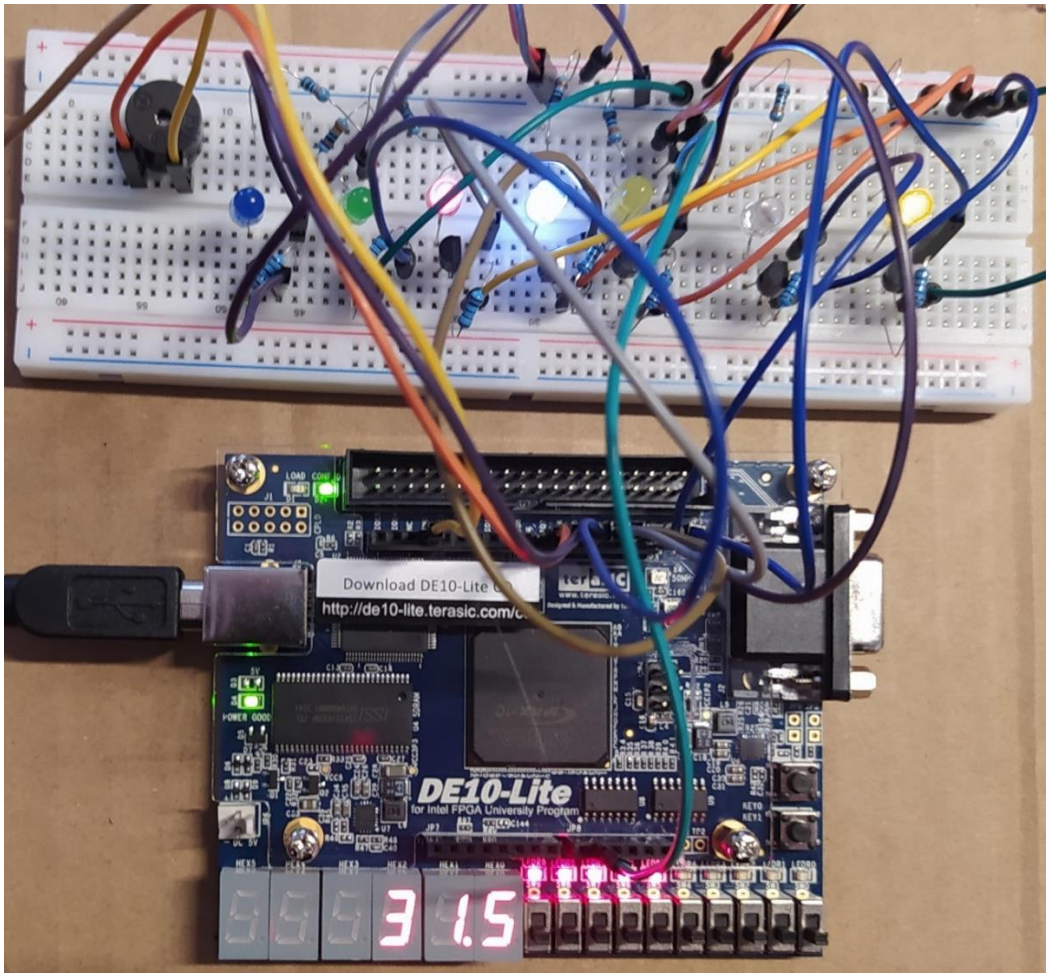

Figure 5: The multiple temperature-control system of this work, operating. Temperature value is above upper set limit and temperature rate decrease exceeds set value. Temperature values remained higher than upper set limit for specific time set period.

III. PROGRAMING THE SYSTEM

We used Quartus Prime Lite Edition 21.1.1 to create the VHDL programs of our system. It must be mentioned here that before proceeding with the VHDL programming of our system, we had to set a series of parameters controlling the operation of DE10-Lite FPGA's Analog to Digital Converter (ADC). This converter plays a very important role in the whole system operation, since it converts the analogue input voltage from LM35 temperature sensor connected to FPGA board to digital values, acting as main input of the system. The files created by the above ADC parameters setting are imported into the final project of our system.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

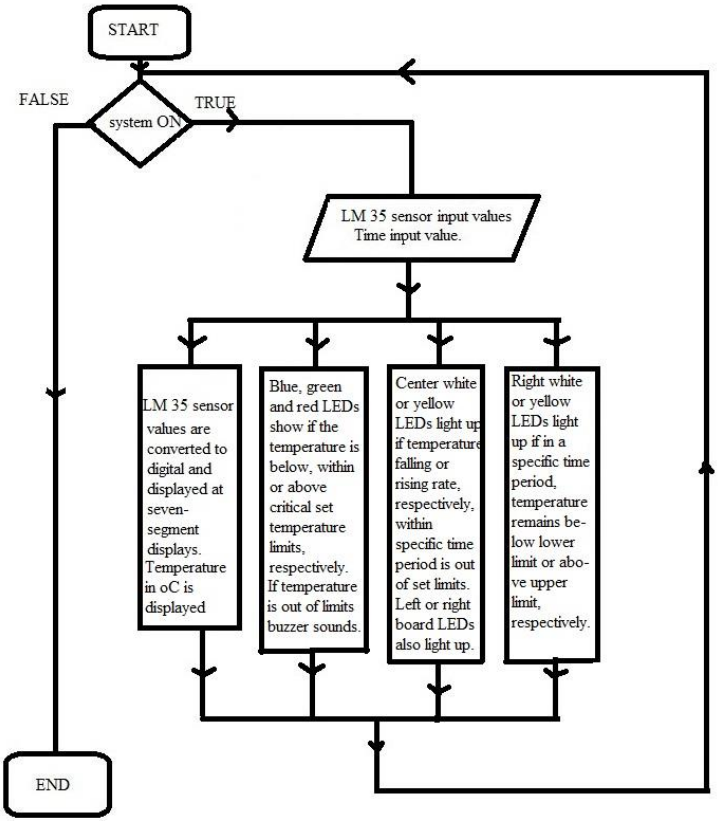

Figure 6: Flowchart diagram, presenting main functions-processes of our system.

A flowchart diagram, presenting main functions of our system is presented in Figure 6, while the APPENDIX contains the whole VHDL program.

It is clear that the system basically operates four functions. All of them use processes in VHDL programming language. The first function playing definitive role in system operation, uses analog input voltages provided from LM35 temperature sensor, shown in Figure2, as main input, convert them to digital values and present the final result in seven-segment displays as temperature value in Celsius degrees, with an accuracy of one decimal place. These processes are running as long as the system is ON. Calculated values for input sensor voltage values are used in many other processes, in order to activate control systems and external or internal LEDs. Second function includes such processes which activate temperature monitoring LED systems, in order to present the scale of monitored temperature by lighting corresponding LED. In addition to this process we also programmed external buzzer system to sound, whenever temperature is lower than set limit of 18^{o}C or higher than upper set limit of 30^{o}C. Needless to say that our program gives the ability of setting temperature limits according to the application that our system is used.

Third function of our system deals with monitoring increase or decrease rate of temperature. This means that programmer must set a specific difference between two LM35 sensor input voltages Vr1 and Vr2 which correspond to specific time values in seconds. Vr1 is the input voltage value at 30sec and Vr2 is the input voltage value at 60sec. Time values can be set by the programmer depending on the application that our system is used. Taking into account that Vr =10 corresponds to 0.1V input and is converted to a temperature value of 1^{o}C and Vr =100 corresponds to 1V and is converted to a temperature value of 10^{o}C, as we present in the VHDL code in the APPENDIX, we can conclude that (Vr2-Vr1)/(60-30) >1 gives an increase rate of 3^{o}C in 30sec time, while (Vr1-Vr2)/(60-30) >1 gives a decrease rate of 3^{o}C in 30sec time. In the first case center yellow LED is programmed to light up with simultaneous right half of board LEDs lighting, while in the second case both center white LED and left half of board LEDs light up.

Fourth basic function of our system controls the unit responsible for monitoring and counting in a specific time period, the number of times that input temperature is lower than minimum set value or higher than maximum set value. It uses a counter system controlled by FPGA's board clock, so it is the programmer who decides the critical number of times to be exceeded in order to activate corresponding LED. As we mentioned earlier, external white LED, at the right of

DOI: 10.17148/IJIREEICE.2025.131018

breadboard, will be activated if input temperature remains lower than minimum set value for critical clock times set by programmer, while yellow LED will light up in case that temperature values remain higher than maximum set value.

IV. CONCLUSION

A novel FPGA-based system is presented here, which manages to monitor temperature values of a given space or human and activate corresponding control systems, in our case internal or external LEDs and buzzer system. The system uses the LM35 temperature sensor and it is capable of simultaneous monitoring three basic temperature-related procedures. Temperature range of the monitored object, as well as temperature increase or decrease rate in a specific time period and temperature remaining below lower set value or above upper set value for a specific time period. Time periods, lower and upper temperature values and increase or decrease rates are simultaneously monitored and can be set to values that each application requires. This gives to our system the advantage of implementation to a variety of applications. The system can work with all commercial temperature sensors that provide an analog output. Our system is easy to be manufactured, providing also the benefit of low cost.

REFERENCES

[1]. S. Velusamy; W. Huang; J. Lach; M. Stan; K. Skadron, "Monitoring temperature in FPGA based SoCs", International Conference on Computer Design, October 2005, IEEE,ISBN:0-7695-2451-6 DOI: 10.1109/ICCD.2005.78

[2]. T. N. Kumar, H.F. Mohamed, M. Naleem, V. Ganeish, "An FPGA based real-time remote temperature measurement system", International Conference on Electronic Devices, Systems and Applications, 2010, DOI:10.1109/ICEDSA.2010.5503070

[3]. K. Mahant, A.V. Patel, A.Val, R. Goswami, "FPGA based temperature control and monitoring system for X-ray measurement instrument", IEEE Conference (TENCON), 2016, DOI:10.1109/TENCON.2016.7848651

[4]. Y. Xu, "FPGA Alarm System Based on Multi Temperature Sensor", International Journal of Online and Biomedical Engineering (iJOE), 13(05), pp. 109–121, 2017, DOI: https://doi.org/10.3991/ijoe.v13i05.7053

[5]. R. A. Téllez, M. G.-Bosque, G. D.-Señorans, S. Celma, "Temperature Sensors in FPGA Based On Digital Nonlinear Oscillators for Improved Sensitivity", 21st International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuits Design (SMACD), July 2025, DOI:10.1109/SMACD65553.2025.11092156

APPENDIX

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_signed.ALL;
use ieee.std_logic_unsigned.ALL;

entity DE10_Lite_ADC_LM35 is
generic(ClockFrequencyHz : integer:=50000000);

port
(
rst : in std_logic;
nRst    : in std_logic; -- Negative reset
Seconds : inout integer;
Ticks : inout integer;
led1: out std_logic;--1-5(0-4) LEDs light up with large rate of temperature increase
led2: out std_logic;
led3: out std_logic;
led4: out std_logic;
led5: out std_logic;
led6: out std_logic;--6-10(5-9) LEDs light up with large rate of temperature decrease
led7: out std_logic;
led8: out std_logic;
led9: out std_logic;
led10: out std_logic;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

```
led_blue: buffer std_logic; --temperature below lower limit
led_red: buffer std_logic;--temperature above upper limit
led_green: buffer std_logic;--normal
led_blue_out: out std_logic;--temperature below lower limit
led_red_out: out std_logic;--temperature above upper limit
led_green_out: out std_logic;--normal

buzzer:out std_logic; --rings on temperature higher or lower than critical limits
Vr: buffer integer;

d2bbuf :buffer integer range 0 to 9;
d1bbuf :buffer integer range 0 to 9;
d0bbuf :buffer integer range 0 to 9;
SW0 : in std_logic;

yellow_led_alarm_temp_rate: out std_logic;--temperature increase rate out of limit
white_led_alarm_temp_rate: out std_logic;--temperature decrease rate out of limit
yellow_led_alarm_times_temp_higher: out std_logic;--temperature higher than upper limit value many times (out of
limit)
white_led_alarm_times_temp_lower: out std_logic;--temperature lower than lower limit value many times (out of limit)
yellow_led_alarm_temp_rate_buff: buffer std_logic;----temperature increase rate out of limit
white_led_alarm_temp_rate_buff: buffer std_logic;--temperature decrease rate out of limit
yellow_led_alarm_times_temp_higher_buff: buffer std_logic;--temperature higher than upper limit value many times
(out of limit)
white_led_alarm_times_temp_lower_buff: buffer std_logic;--temperature lower than lower limit value many times (out
of limit)

-- Clocks
ADC_CLK_10: in std_logic;
MAX10_CLK1_50: in std_logic;
MAX10_CLK2_50: in std_logic;
-- KEYs
KEY: in std_logic_vector(1 downto 0);

-- HEX
HEX0: out std_logic_vector(7 downto 0);
HEX1: out std_logic_vector(7 downto 0);
HEX2: out std_logic_vector(7 downto 0);


ARDUINO_IO: inout std_logic_vector(15 downto 0);
ARDUINO_RESET_N: inout std_logic);

-- GPIO
--GPIO: inout std_logic_vector(35 downto 0));
end entity;
architecture DE10_Lite_ADC_LM35_Arch of DE10_Lite_ADC_LM35 is

-- Analog to Digital Converter IP core
component myADC is
port
(
clk_clk: in std_logic := 'X';
modular_adc_0_command_valid: in std_logic := 'X';
modular_adc_0_command_channel: in std_logic_vector(4 downto 0) := (others => 'X');
modular_adc_0_command_startofpacket: in std_logic := 'X';
modular_adc_0_command_endofpacket: in std_logic := 'X';
modular_adc_0_command_ready: out std_logic;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131018

```
modular_adc_0_response_valid: out std_logic;
modular_adc_0_response_channel: out std_logic_vector(4 downto 0);
modular_adc_0_response_data: out std_logic_vector(11 downto 0);
modular_adc_0_response_startofpacket: out std_logic;
modular_adc_0_response_endofpacket: out std_logic;
reset_reset_n: in std_logic
);
end component myADC;
signal modular_adc_0_command_valid: std_logic;
signal modular_adc_0_command_channel: std_logic_vector(4 downto 0);
signal modular_adc_0_command_startofpacket: std_logic;
signal modular_adc_0_command_endofpacket: std_logic;
signal modular_adc_0_command_ready: std_logic;
signal modular_adc_0_response_valid: std_logic;
signal modular_adc_0_response_channel: std_logic_vector(4 downto 0);
signal modular_adc_0_response_data: std_logic_vector(11 downto 0);
signal modular_adc_0_response_startofpacket: std_logic;
signal modular_adc_0_response_endofpacket: std_logic;
signal clk_clk: std_logic;
signal reset_reset_n: std_logic;
type state_machines is (sm0,sm1, sm2, sm3, sm4);
signal sm: state_machines;
-- signals to store conversion results
signal ADCIN1,ADCIN4, ADCIN3,ADCIN2: std_logic_vector(11 downto 0);
signal AD1,AD4, AD3,AD2: std_logic_vector(11 downto 0);
-- signal for BCD digits
signal digit2b, digit1b, digit0b: std_logic_vector(3 downto 0);
signal digit2, digit1, digit0: std_logic_vector(3 downto 0);

-- signal to determine how fast the
-- 7-seg displays will be updated
signal cnt: integer;
signal state_LED_right: std_logic;
signal state_LED_left: std_logic;
signal state_Vr: integer;
signal xmax_maximum: integer;
signal xmin_minimum: integer;

begin
-- ADC port map
adc1: myADC port map
(
modular_adc_0_command_valid => modular_adc_0_command_valid,
modular_adc_0_command_channel => modular_adc_0_command_channel,
modular_adc_0_command_startofpacket => modular_adc_0_command_startofpacket,
modular_adc_0_command_endofpacket => modular_adc_0_command_endofpacket,
modular_adc_0_command_ready => modular_adc_0_command_ready,
modular_adc_0_response_valid => modular_adc_0_response_valid,
modular_adc_0_response_channel => modular_adc_0_response_channel,
modular_adc_0_response_data => modular_adc_0_response_data,
modular_adc_0_response_startofpacket => modular_adc_0_response_startofpacket,
modular_adc_0_response_endofpacket => modular_adc_0_response_endofpacket,
clk_clk => clk_clk,
reset_reset_n => reset_reset_n
);
clk_clk <= MAX10_CLK1_50;
reset_reset_n <= KEY(0);
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

```
-- process for reading new samples
p1: process(reset_reset_n, clk_clk)
begin
if reset_reset_n = '0' then
        sm <= sm0;
elsif rising_edge(clk_clk) then
        case sm is
                when sm0 =>
                        sm <= sm1;
                        modular_adc_0_command_valid <= '1';
                        modular_adc_0_command_channel <= "00001";
                when sm1 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00010";
                         ADCIN4 <= modular_adc_0_response_data;
                                sm <= sm2;
                        end if;
                when sm2 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
                                modular_adc_0_command_channel <= "00011";
                                ADCIN1 <= modular_adc_0_response_data;
                                sm <= sm1;
                                sm <= sm3;
                        end if;
                when sm3 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00100";
                                ADCIN2 <= modular_adc_0_response_data;
                                sm <= sm4;
                        end if;
                when sm4 =>
                        if modular_adc_0_response_valid = '1' then
                                modular_adc_0_command_channel <= "00001";
                                ADCIN3 <= modular_adc_0_response_data;
                                sm <= sm1;
                        end if;
                when others =>
        end case;
end if;
end process;

-- process for conversion from binary to BCD (analog input voltage)
p3: process(AD2,d2bbuf,d1bbuf,d0bbuf)
variable vin: integer;
variable d2, d1, d0: integer;
begin

vin := to_integer(unsigned(std_logic_vector(to_unsigned(to_integer(unsigned(AD2)) * 500,
32))(31 downto 12)));
d2 := vin / 100;
d1 := vin mod 100 / 10;
d0 := ((vin mod 100) mod 10);
digit2b <= std_logic_vector(to_unsigned(d2, 4));
digit1b <= std_logic_vector(to_unsigned(d1, 4));
digit0b <= std_logic_vector(to_unsigned(d0, 4));
d2bbuf<= d2;
d1bbuf<= d1;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

```
d0bbuf<= d0;

end process;

state_Vr<= (d2bbuf*100)+(d1bbuf*10)+(d0bbuf);
Vr<= state_Vr;

-- determine how fast the 7-seg displays will be updated
p4: process(reset_reset_n, clk_clk)
begin
if reset_reset_n = '0' then
cnt <= 0;
elsif rising_edge(clk_clk) then
if cnt < 20_000_000 then
cnt <= cnt + 1;
else
cnt <= 0;
AD1 <= ADCIN1;
AD2 <= ADCIN2;
AD3 <= ADCIN3;
AD4 <= ADCIN4;
end if;
end if;
end process;

--time-seconds
process(MAX10_CLK1_50) is
   begin
      if rising_edge(MAX10_CLK1_50) then --ισοδύναμο του IF CLK'EVENT AND CLK='1'

         ------ If the negative reset signal is active
         if nRst = '0' then
            Ticks   <= 0;
            Seconds <= 0;
         else
            -- True once every second
            if Ticks = ClockFrequencyHz - 1 then
               Ticks <= 0;
                  Seconds <= Seconds + 1;
            else
               Ticks <= Ticks + 1;
            end if;
         end if;
      end if;
   end process;

process (Vr,seconds,MAX10_CLK1_50, yellow_led_alarm_temp_rate_buff,
white_led_alarm_temp_rate_buff, yellow_led_alarm_times_temp_higher_buff,
white_led_alarm_times_temp_lower_buff, state_LED_right,state_LED_left,
xmax_maximum, xmin_minimum)

variable Vr1: integer;
variable Vr2:integer;

begin

IF  seconds=30 THEN--50000000 Ticks= 1sec and Vr=10 corresponds to 0.1V input and 1oC and 100 to 1V and 10oC
Vr1:=Vr;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131018

```
end if;
IF  seconds=60 THEN--50000000 Ticks= 1sec and Vr=10 corresponds to 0.1V input 1oC and 100 to 1V and 10oC
Vr2:=Vr;
end if;
IF ((Vr2-Vr1)/(60-30))>1 THEN --3oC increase in 30 seconds
yellow_led_alarm_temp_rate_buff<='1';
state_LED_right<='1';
else
yellow_led_alarm_temp_rate_buff<='0';
state_LED_right<='0';
end if;
IF ((Vr1-Vr2)/(60-30))>1 THEN --3oC decrease in 30 seconds
white_led_alarm_temp_rate_buff<='1';
state_LED_left<='1';
else
white_led_alarm_temp_rate_buff<='0';
state_LED_left<='0';
end if;

if MAX10_CLK1_50'event and MAX10_CLK1_50 = '1' then
IF ((seconds>=120 AND seconds<=240) AND Vr>300) THEN
xmax_maximum<=xmax_maximum+1;
else
xmax_maximum<=xmax_maximum;
end if;
end if;

IF (xmax_maximum>=3) THEN
yellow_led_alarm_times_temp_higher_buff<='1';
else
yellow_led_alarm_times_temp_higher_buff<='0';
end if;

if MAX10_CLK1_50'event and MAX10_CLK1_50 = '1' then
IF ((seconds>=120 AND seconds<=240) AND Vr<180) THEN
xmin_minimum<=xmin_minimum+1;
else
xmin_minimum<=xmin_minimum;
end if;
end if;

IF (xmin_minimum>=3) THEN
white_led_alarm_times_temp_lower_buff<='1';
else
white_led_alarm_times_temp_lower_buff<='0';
end if;
end process;

yellow_led_alarm_temp_rate<=yellow_led_alarm_temp_rate_buff;
white_led_alarm_temp_rate<=white_led_alarm_temp_rate_buff;
yellow_led_alarm_times_temp_higher<=yellow_led_alarm_times_temp_higher_buff;
white_led_alarm_times_temp_lower<=white_led_alarm_times_temp_lower_buff;

led1<=state_LED_right;
led2<=state_LED_right;
led3<=state_LED_right;
led4<=state_LED_right;
led5<=state_LED_right;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131018

```
led6<=state_LED_left;
led7<=state_LED_left;
led8<=state_LED_left;
led9<=state_LED_left;
led10<=state_LED_left;

process (Vr, led_blue,led_red, led_green)
begin
IF Vr>300 THEN
led_red<='1';
else
led_red<='0';
end if;

IF Vr<180 THEN
led_blue<='1';
else
led_blue<='0';
end if;

IF Vr<=300 AND Vr>=180 THEN
led_green<='1';
else
led_green<='0';
end if;
end process;

led_blue_out<=led_blue;
led_red_out<=led_red;
led_green_out<=led_green;

        --critical pulses_per_minute value exceeded buzzer sounds
Process(MAX10_CLK1_50,Vr)
variable i : integer := 0;
BEGIN
IF (Vr>300) OR (Vr<180) THEN --temperature >30oC OR temperature<18oC buzzer sounds
if MAX10_CLK1_50'event and MAX10_CLK1_50 = '1' then
if i <= 50000000 then
i := i + 1;
buzzer <= '1';
elsif i > 50000000 and i < 100000000 then
i := i + 1;
buzzer <= '0';
elsif i = 100000000 then
i := 0;
end if;
end if;
end if;
end process;

process(SW0,digit2b,digit1b,digit0b)
begin
  IF SW0='0' THEN
  digit2 <= digit2b;--first digit of temperature (tens)
  digit1 <= digit1b;--second digit of temperature (units)
  digit0 <= digit0b;--third digit of temperature (10^-1)
  end if;
end process;
```

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131018

```
 WITH digit2 SELECT
HEX2 <= "01000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit1 SELECT
HEX1 <= "11000000" WHEN "0000", -- display 0
"01111001" WHEN "0001", -- display 1
"00100100" WHEN "0010", -- display 2
"00110000" WHEN "0011", -- display 3
"00011001" WHEN "0100", -- display 4
"00010010" WHEN "0101", -- display 5
"00000011" WHEN "0110", -- display 6
"01111000" WHEN "0111", -- display 7
"00000000" WHEN "1000", -- display 8
"00011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display

WITH digit0 SELECT
HEX0 <= "11000000" WHEN "0000", -- display 0
"11111001" WHEN "0001", -- display 1
"10100100" WHEN "0010", -- display 2
"10110000" WHEN "0011", -- display 3
"10011001" WHEN "0100", -- display 4
"10010010" WHEN "0101", -- display 5
"10000011" WHEN "0110", -- display 6
"11111000" WHEN "0111", -- display 7
"10000000" WHEN "1000", -- display 8
"10011000" WHEN "1001", -- display 9
"11111111" WHEN OTHERS; -- blank display
end architecture;
```