THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2021, VOL. 44, NO. 1, ISSN 2344-4738, ISSN-l 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS
Article DOI: https://doi.org/10.35219/eeaci.2021.1.01

ARDUINO TEMPERATURE MONITORING SOLUTION FOR ROUTERS

Laurentiu Marius Baicu, Bogdan Dumitrascu,
Mihaela Andrei

"Dunarea de Jos" University, Domneasca Street no. 47, 800008 Galati, Romania
Department of Electronics and Telecommunications
laurentiu.baicu@ugal.ro, bogdan.dumitrascu@ugal.ro, mihaela.andrei@ugal.ro

Abstract: An important element in communications systems is the router. The paper proposes a remote temperature monitoring system for it. This was implemented based on Arduino Uno development platform. In addition, an Android user-friendly interface has been created; it can be installed on a smart terminal (phone or tablet) and its role is to display the router temperature and to notify the user if it has exceeded a maximum set value. There are two methods of notifications: one is using different colors for the displayed temperatures and the other using an alert. The temperature is measured with a sensor and the data is then processed by the microcontroller and sent to the Bluetooth module using serial interface. Also, the measured values can be seen on an LCD display. The hardware, software designs and the simulation and experimental results of the proposed system are presented.

Keywords: router; temperature monitoring; mobile application; Bluetooth; Arduino.

1. INTRODUCTION

Digitization is present in all activity-sectors as well as fast and secure communications that have made people's lives much easier. A key element in communications systems used for network interconnection is the router because it improves the access to information. This device operates on Network and Transport Layers in OSI model [15], its functioning being based on packet switching. There are many types of routers such us edge, core, or virtual ones [17, 3], some of them leading to high performances through the transfer of many packets in very little time [4].

Its role is very important because the Internet of Things (IoT) applications are so popular nowadays, a router being responsible with security systems used in IoT [2]. In addition, to the software problems that may occur with a router [17], its overheating represents an essential aspect. Until now, this was not treated in literature, although there are some monitoring software solutions: PRTG Network Monitor [8], Manage Engine OpManager [7], Zenos CiscoMonitor ZenPack [10] or WhatsUp Gold Network Traffic Monitor [9].

Considering the aforementioned issue, the present paper proposes a router temperature monitoring system based on Arduino Uno platform and a user-friendly Android application which can be installed on a smart terminal: a phone or a tablet with Android. The temperature is measured with a LM35 sensor and displayed on a Liquid Crystal Display (LCD) and on the smartphone screen through the interface created in MIT APP inventor. The application role is to announce the user that the router temperature is high and that is made with two different types of notifications: with colors and with a sound alert.

The Arduino platform is very used in many projects and applications due to its advantages such us: portability, low price, open-source libraries, integrated hardware and software, sensors compatibility etc. [12, 5]. The temperature monitoring system based on Arduino was designed also for retail refrigerated cabinets [14], for medical use [13] or for a lot of IoT applications [1, 16, 11]. Since IoT solutions can be very useful, especially in emergencies, but they require an internet connection to properly function, it is very important to ensure a stable internet connection. The most popular way to ensure an internet connection for homes is the usage of routers to connect to the Internet Service Provider and all the IoT solutions in the house are connected to it. If the router has a malfunction, the connection to the internet is lost and all the remote functions of the IoT solutions become useless.

To make it easier to monitor and control the abovementioned systems, Android interfaces have been created [16]. In this paper the software used for it was MIT APP Inventor.

The rest of the paper is organized as follows: in section 2 it is presented the hardware design and implementation, in section 3 the software part. The section 4 presents the simulation and experimental results, and the section 5 concludes the paper.

2. SYSTEM DESIGN

The proposed system design is represented in Fig. 1. It is developed with a board designed by Arduino named Arduino Uno, a Bluetooth module, a Texas Instruments LM35 sensor temperature and an LCD display. The platform is built with an Atmel microcontroller: Atmega 328P, very popular among electronic developers [12]. It has 28 pins divided into ports and grouped into different type of categories. There are 14 input/output pins, 6 of them being able to be used as Pulse-Width-Modulation (PWM) outputs and other 6 pins that can be used as analog inputs. The rest of 8 pins are mainly power pins and analog reference pin. Beside the microcontroller the board also has an In-Circuit Serial Programming (ICSP) header, reset button, external power connector, 16MHz external crystal oscillator and USB support that can be used for programming the microcontroller or simply for powering the board. This support feature is possible thanks to another microcontroller named Arduino 8U2 that is programmed as a USB to serial converter, also present on the board.

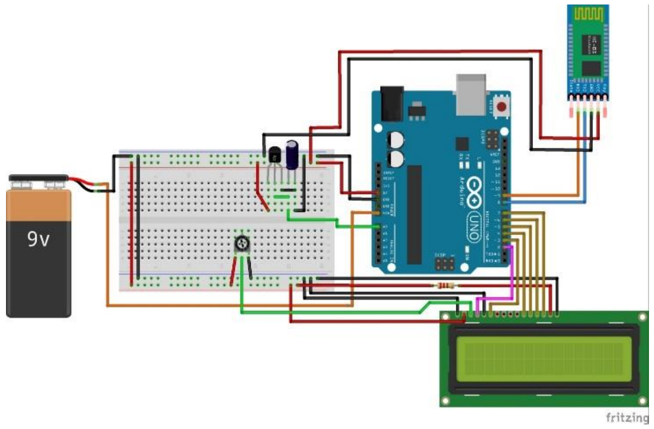

Fig. 1. Fritzing schematic of the monitoring system

The board can be powered via USB connector using 5 volts or using external power supply that can be connected using a 2.1mm center positive plug into the board power jack. The power jack also supports a 9 volt battery but it needs to have an adapter for the power jack connector. Another method for powering the board is by using two wires from a battery having the + one connected in the V_{in} pin like in the block diagram Fig.1. If non-USB powering method is chosen, it is recommended to be used a voltage between 7 to 12 volts. Using less than 7V, the board may become unstable and using more than 12V, the board will start to overheat because of the voltage regulator. If external power in this range is used on the power jack or on the Vin pin, the board voltage regulator will convert the power to 5 volts which can be then outputted via 5V pin or via 3.3 volts pin used for low power modules.

The temperature is measured with a Texas Instruments LM 35 precision temperature sensor. It can measure temperatures between -55 degrees Celsius to 150 degrees Celsius. This type of sensor was used over the more precise linear sensors calibrated in Kelvin scale because this sensor is already calibrated in Centigrade (Celsius) and does not require any software adjustments to provide the needed accuracy for this application, around 1/4˚C.

The sensor can be supplied with voltages from the following range 4V to 20V. Using the board regulated power supply, the sensor will be powered with 5V. In this range, the sensor will output a signal of 0 to 10mV for 1 degree Celsius.

To display the measured values an LCD module was used. This is a 1602 display using parallel communication with external electronics, in this case the microcontroller. The module is driven by a Hitachi HD44780U controlled and it can display 2 lines and 16 characters (squares of pixels). Also, the module has backlight illumination that can be adjusted using a potentiometer or it can also be turned off for less power consumption, the current used only by the display being around 1mA. The LCD display is powered using the same regulated voltage as the development board and a resistor for protection. Considering that the sensor will measure the temperature at the surface of radiators of telecommunications equipment, we aim for a range of temperatures between 30˚C to 80˚C, so the sensor measuring capabilities are in the range of our application requirements. For noise filtering an electrolytic capacitor is used on the signal output of the temperature sensor.

3. SOFTWARE DESIGN

This section has two parts: one is for Arduino software and one for Android application.

3.1 Arduino Software

In order to transmit the measured values from the sensor to the smartphone application using Arduino Uno and a HC05 Bluetooth module, a software algorithm needs to be implemented on the Arduino Uno module. So, Arduino collects data from the temperature sensor and it uses an average of 100 readings to calculate the temperature in degrees Celsius and Fahrenheit and it displays the temperature on the LCD display and also sends it to the Bluetooth module using the mySerial.print function. The algorithm schematic is presented in Fig. 2.

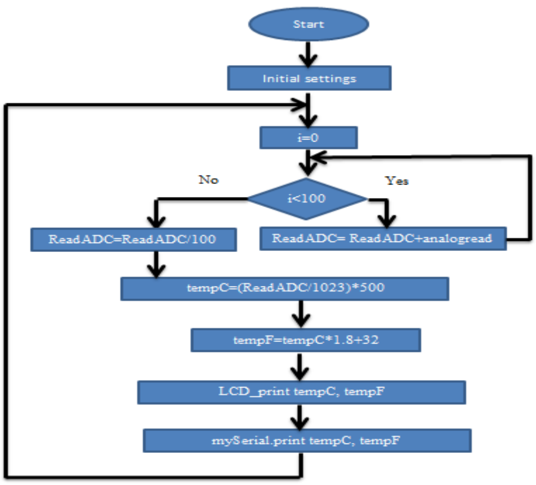

Fig. 2. Schematic of the Arduino software

The initial settings will define the input pin for temperature readings, output pins for the communications with the Bluetooth module and LCD display and the declaration of variables for analogue read, temperature and initialization with 0. In the infinite loop, the temperature is calculated and sent to the display and Bluetooth. In order to avoid measurement noise, the average of 100 values received from the analogue sensor and converted using analogRead() function is used when calculating the temperature. The temperature in degrees Celsius is obtained using:

$$tempC = (\mathrm{Re}\,adADC\,/1023)\cdot 500° \quad (1)$$

where: -tempC is the temperature in degrees Celsius

-ReadADC is the average of the sensor data.

The temperature in degrees Fahrenheit is calculated using:

$$tempF = tempC \cdot 1.8 + 32\,. \quad (2)$$

2.2 Mobile Application

We developed a user-friendly mobile application in MIT APP Inventor. This software was created by Google Labs in 2008. The application can be installed on a smart terminal, a phone or a tablet, with Android System. The communication with Arduino is realized via Bluetooth (BT). The screen is represented in Fig. 3.

Fig. 3. Mobile application

The application has a button to connect and one to disconnect the phone to and from the designed system. The BT module will be selected from a list and in the next step the measured temperature will be displayed in Celsius and Fahrenheit. The important role of this application is to announce the user when the router temperature is high. Three levels are set and marked with different colors: green for temperatures lower than 50°C, orange for temperatures between 50°C and 60°C and red when it is higher than 60°C. In that moment, there is another type of notification: sound alert.

4. RESULTS

In order to validate the proposed solution, a series of experiments were conducted consisting of simulations and platform testing results.

5.1 Simulation results

For the proposed system, a simulation was made using Labcenter Proteus software. This software allows the simulation of the software code written in the microcontroller and also to adjust different values/parameters that needed adjustment before switching to the real testing of the system. The schematic is presented in Fig. 4 and the components are described in section 2: the Arduino board, the LCD display, the temperature sensor, the Bluetooth module. The Bluetooth module has a virtual terminal connected so it can simulate the sending or the receiving data via USART interface.

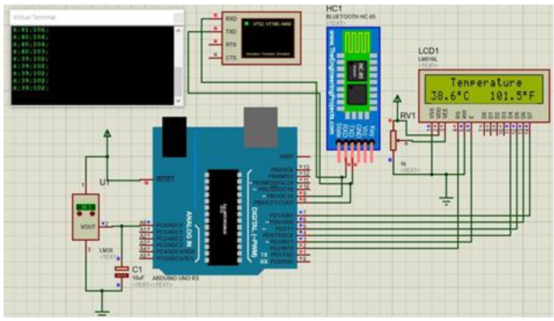

Fig. 4. Proteus schematic and simulation

In Fig. 5 there are presented the simulation results. After uploading the required code in the microcontroller memory and running the simulation in the left monitor, the data that is sent via USART interface to the Bluetooth module can be seen. Also, the data can be seen on the LCD display.

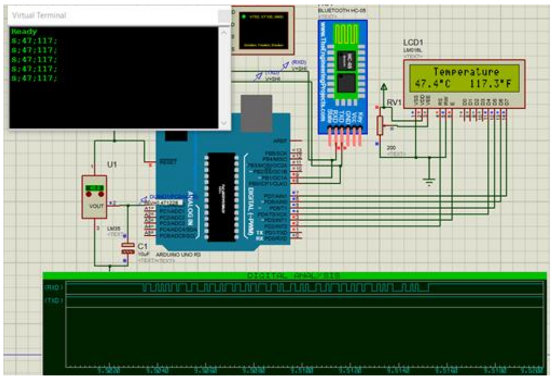

Fig. 5. Simulation done in Proteus

During the test, there was also used an advanced simulation feature such us a digital analysis feature. In Fig. 6 it is displayed the UART signal.

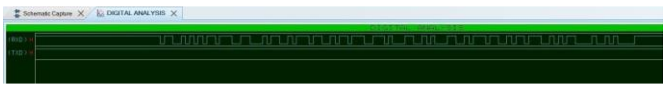

Fig. 6. USART signal

The signal shown above represents the "47" degrees value that is sent to the Bluetooth module. For this, probes were connected to the RX pin. This can be seen in Fig.6.

5.2 Experimental results

For the proposed system we have done real testing to determine the error of measuring. The system was tested on a gigabit ASUS RT-AC1200G+ router [6]. The router is an AC standard router which has also an USB port and it can work as a server if the USB port is used for connecting an SSD or Hard Drive. Also having the AC standard means that it can work wireless on the standard frequency of 2,4Ghz but also on the 5Ghz band which generates more heat into the router board. The device has a 900MHz Broadcom BCM47189 CPU and 128 MB of NANYA RAM memory and can reach maximum speeds of 300 Mbps on 2,4GHz band and 867 Mbps on 5GHz band.

The sensor was mounted on the router's main radiator (Fig.7) where the CPU router is also located, being one of the parts that generates the most heat. It is essential that this type of performance routers do not get temperatures over 60 degrees C which can lead to solder balls cracking and making the device unusable.

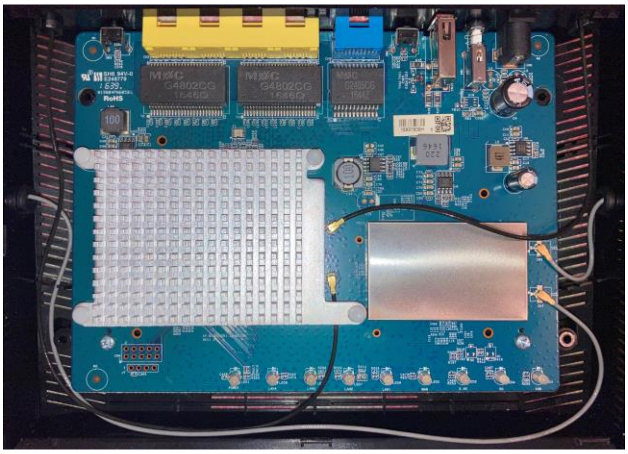

Fig. 7. Router radiator

Our proposed system can determine the state of thermal pads in time if the router temperatures are starting to get high in idle. Thermal pads are used to transfer heat from the chips to the radiators, but they can deteriorate overtime and lead to higher temperatures and higher degradation of the board. Since devices like these do not have a temperature monitor not even in software, it is almost impossible to determine the thermal material condition without looking at them periodically, so a monitoring system can be very convenient to use. Changing the router thermal pads when it is needed can extend the life of the routers board.

The monitoring system can also be used to see which load generates more heat to the circuit board. In our test, we have compared the temperature measured by the sensor with an external infrared thermometer.

The first measured temperatures were read using the router in idle mode at only 2.4 GHz and then at 5 GHz (Fig. 9).

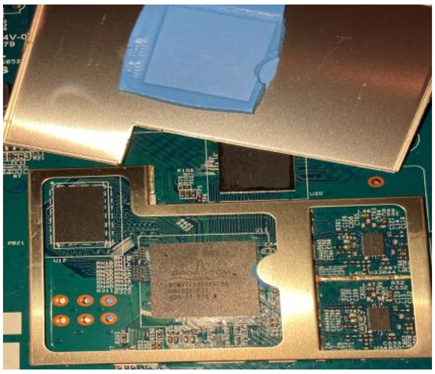

Fig. 8. Router Broadcom CPU, memory chip and thermal pad

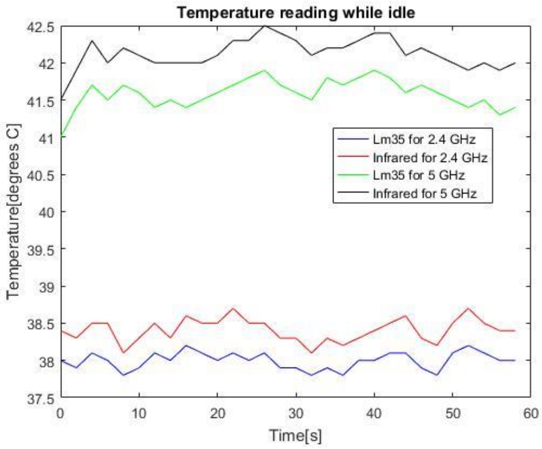

Fig. 9. Idle temperatures 2.4 GHz and 5GHz

The second case presents values read with the router under load stress also at 2,4GHz and 5GHz (Fig. 10). Load stress was generated transferring big data files via wifi on external SSD connected to router.

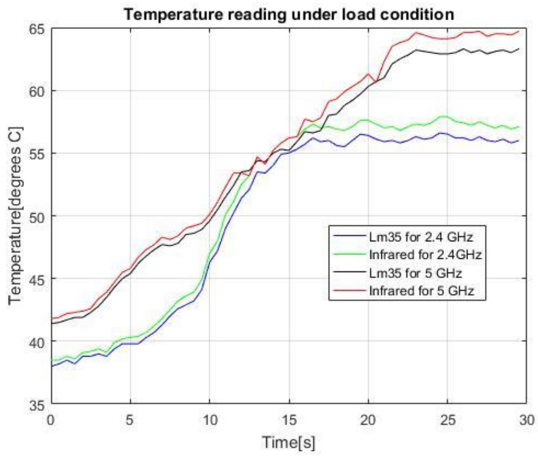

Fig. 10. Load stress temperatures 2.4 GHz and 5GHz

In Fig.11 and Fig.12 there are represented the differences between the temperature measured with LM35 sensor and an external infrared sensor. From the graphs with the measurement error, we can see that the error of the LM35 sensor becomes higher with increasing temperature.

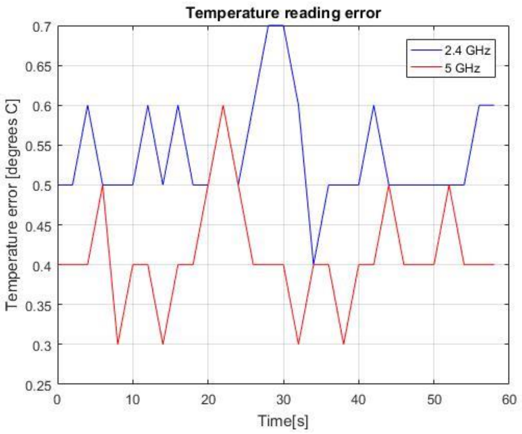

Fig. 11. Error for idle measured temperatures

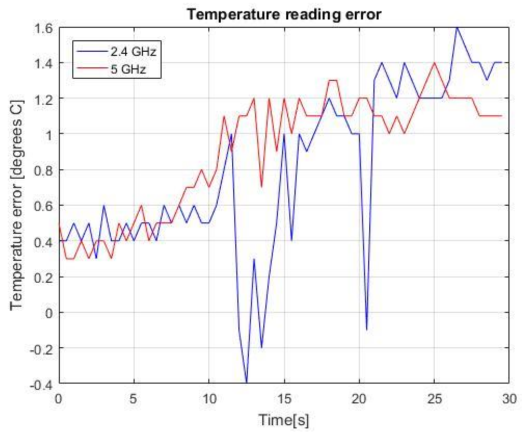

Fig. 12. Error for load measured temperatures

5. CONCLUSIONS

A solution for monitoring the temperature of a router involving a temperature sensor, an Arduino Uno and a mobile application is proposed. The user can decide based on the temperature data collected if the router cooling system works normally or the thermal pads need to be changed. Futhermore, there can be measured different type of load stress to the router. Future work may involve upgrading the temperature monitoring system with an active cooling solution to improve cooling furthermore and the ability to take decisions based on the read data.

REFERENCES

[1]. A-hussein, A., Salman, A. (2020), IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol, *Iraqi Journal of Computers, Communications, Control & Systems Engineering*, 20(3), DOI: https://doi.org/10.33103/uot.ijccce.20.3.7

[2]. Alanezi, M.A., Bouchekara, H.R.E.H., Javaid, M.S. (2020). Optimizing Router Placement of Indoor Wireless Sensor Networks in Smart Buildings for IoT Applications. *Sensors*, *20*, 6212. https://doi.org/10.3390/s20216212

[3]. Beshley, M., Kryvinska, N., Beshley, H.; Yaremko, O.; Pyrih, (2021). J.Virtual Router Design and Modelingfor Future Networks with QoSGuarantees. *Electronics*, 10, 1139, https://doi.org/10.3390/electronics10101139

[4]. Fernández, C., Giménez, S., Grasa, E., Bunch, S. (2020)A P4-Enabled RINA Interior Router for Software-Defined Data Centers. *Computers*, *9*, 70. https://doi.org/10.3390/computers9030070

[5]. Gao, W., Luo, X., Liu, Y., Zhao, Y., Cui, Y. (2021). Development of an arduino-based integrated system for sensing of hydrogen peroxide, *Sensors and actuators reports*, 3, 100045, https://doi.org/10.1016/j.snr.2021.100045.

[6]. https://www.asus.com/ro/Networking-IoT-Servers/WiFi-Routers/ASUS-WiFi-Routers/RT-AC1200G-plus/

[7]. https://www.manageengine.com/network-monitoring/

[8]. https://www.paessler.com/

[9]. https://www.whatsupgold.com/network-monitoring-software

[10]. https://www.zenoss.com

[11]. Islam, M.M., Kashem, M.A., and Uddin, J. (2022), An internet of things framework for real-time aquatic environment monitoring using an Arduino and sensors, *International Journal of Electrical and Computer Engineering*, 12(1), pp. 826-833, DOI: 10.11591/ijece.v12i1.pp826-833

[12]. Organtini, G. (2018). Arduino as a tool for physics experiments, *Journal of Physics: Conference Series*, 1076, doi:10.1088/1742-6596/1076/1/012026

[13]. Rahimoon, A., Abdullah, M.N., and taib, I. (2020). Design of a contactless body temperature measurement systems using Arduino, *Indonesian Journal of Electrical Engineering and Computer Science*, 19 (3), pp. 1251-1258, DOI: 10.11591/ijeecs.v19.i3.pp1251-1258

[14]. Ramírez-Faz, J., Fernández-Ahumada, L.M., Fernández-Ahumada, E., López-Luque, R. (2020). Monitoring of Temperature in Retail Refrigerated Cabinets Applying IoT Over Open-Source Hardware and Software. *Sensors*, *20*, 846. https://doi.org/10.3390/s20030846

[15]. Sundararajan, A., Chavan, A., Saleem, D., and Sarwat, A. (2018). A Survey of Protocol-Level Challenges and Solutions for Distributed Energy Resource Cyber-Physical Security. *Energies*, 11, 2360, doi:10.3390/en11092360, pp. 1-21.

[16]. Umapathi, N., and Sabbani, S. (2021). An Internet of Things (IoT) based Approach for Real-Time Kitchen Monitoring Using NodeMCU 1.0, Lecture Notes in Electrical Engineering, Springer, 792, pp. 35-43, https://doi.org/10.1007/978-981-16-4625-6_4

[17]. West, J., dean, T., and Andrews,J. (2016). *Network+ Guide to Network*, Cengage Learning, Boston