WMINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION
Federal State State-Financed Educational Institution of High Professional Education
South Ural State University (National Research University)
Faculty of Computational Mathematics and Informatics
Department of System Programming

THESIS IS CHECKED

Reviewer, PhD, Assoc.Prof.

Dep. IMMPI, ChSPU

_____ O.A. Dmitrieva

"10" июня 2016

ACCEPTED FOR THE DEFENSE

Head of the department, Dr. Sci., Prof.

_____ L.B. Sokolinsky

"13" 06 2016

DEVELOPMENT OF THE APPLICATION FOR MONITORING THE TEMPERATURE AT MACHINE ROOM

GRADUATE QUALIFICATION WORK
SUSU–010300.68.2016.13-137-2598.GQW

Supervisor

Cand. Sci., Assoc. Prof.

_____ O.N. Ivanova

Author,
the student of the group VMI-216

_____ A.M. Alkoaerji

Normative control

_____ O.N. Ivanova

"06" июня 2016

Chelyabinsk–2016

MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

Federal State State-Financed Educational Institution of High Professional Education
South Ural State University (National Research University)
Faculty of Computational Mathematics and Informatics
Department of System Programming

APPROVED

Head of the department,
Dr. Sci., Prof.
_____ L.B. Sokolinsky
"09" 02 2016

TASK
of the master graduate qualification work
for the student of the group VMI-216
Ali Mohammed mozan Al.koaerji
in master direction 02.04.02
"Fundamental Informatics and Information Technologies"
(master program "Database Technologies")

1. **The topic** (approved by the order of the rector from 15.04.2016 No. 661)

Development of the Application for monitoring the temperature at The Super-computer center of South Ural State University

2. **The deadline for the completion of the work:** 05.06.2016.
3. **The source data for the work**

3.1. Garsia-Molina H., Ullman J.D., Widom J. Database systems. The complete book. 2nd Edition. – USA, New Jersey: Pearson Education Inc., 2009. – 1241 p.
3.2. Blum R. Arduino programming in 24 hours. – India: Indianapolis, Indiana Sams, 2015. – 419 p.
3.3. Bennett S. UML. – USA, New York: McGraw-Hill, 2005. – 398 p.
3.4. Zak D. Programming with Microsoft Visual Basic 2010. – USA: Cengage Learning, 2011. – 840 p.

4. **The list of the development issues**

4.1. Describe the problem statement of temperature monitoring.
4.2. Study the modern tools of application development for Arduino micro-controller.

4.3. Design circuit diagram, deployment diagram and use case diagram for the application for monitoring the temperature.

4.4. Develop the structure of database for keeping history of temperature changes and current temperature.

4.5. Develop the interface of the client and administrator applications.

4.6. Implement and test the application.

5. Issuance date of the task: 09.02.2016.

Supervisor

PhD, Assoc. Prof. O.N. Ivanova

The task is taken to perform A.M. Al.koaerji

Student: Al-Koaerji A.M.M.

Supervisor: Ivanova O.N., PhD, Assoc. Prof.

Topic: Development of the system for monitoring temperature at machine room

The calendar plan
of the execution of master graduate qualifying work (GQW)

No	Phase	Duration	Deadline	Report	Actual date of execution	Supervisor's signature
1.	Introduction and literature review	1 month	February, 25th	1. Task of the master graduate qualification work 2. Text of Introduction 3. References	25.02.16	
2.	Development of the model, design of the system	1 month	March, 15th	1. Text of chapter 1 (theoretical part).	15.03.16	
3.	Implementation of a system	1 month	April, 15th	1. Software system 2. Text of chapter 2 (implementation part).	15.04.16	
4.	Testing and debugging of the system, experiments	2 weeks	May, 1st	1. Set of tests 2. Text of chapter 3 (experimental part).	01.05.16	
5.	Full text	2 weeks	May, 15th	1. Full text of GQW	15.05.16	
6.	Validation of the text by supervisor	1 week	May, 22nd	1. Electronic version of the GQW text checked by the supervisor	22.05.16	
7.	Normative control	3 days	May, 25th	1. Twisted text of GQW signed by student, supervisor and normative controller	30.05.16	
8.	Proposal defense	1 week	May, 25th – June, 1st	1. Twisted text of GQW signed by student, supervisor and normative controller for the signature of the Head of the Department Head about accepting for the defense 2. A signed review of the supervisor 3. A review of the reviewer, signed and notarized at his place of work 4. Implementation act (if exists) 5. Presentation of the report in PowerPoint	01.06.16	

Table of contents

Acknowledgements

I would like to express thanks to our God to help me for do this system.

I want to thank my deepest thanks and gratitude to PHD, associate professor, O.N. Ivanova, my supervisor for her guidance, support and continuous enthusiasm and encouragement throughout the project is also very grateful and extend my sincere thanks to all staff working in the Ministry of Higher Education and Scientific Research of Iraq for their cooperation.

Finally many thanks to my friends, who knew me the path of the start to get this opportunity, and who helped me to pass the difficult life in the abroad. Last but not least a lot of thanks to my father, mother, brothers and sisters who have helped and given me the support and confidence to pass this opportunity.

I present the fruit of my success, to the spirit of the martyr's my brother "Qasim" and to all the souls of the martyr's heroes of the popular crowd and the Iraqi Army who are defending Iraq and his land against terrorists.

Introduction

Actuality

In the Supercomputer Center, we need a continuous monitoring system for temperature to protect the servers in this room from damage due to high temperature, especially in the summer.

There are many systems, which monitor the temperature in rooms in the market, but none of them can inform the engineers and other staff about problems in their absence. These systems can produce the alarming sound; can also automatically switch on/off the coolers. Simultaneously, they can’t send e-mails or SMS. In addition, it is impossible to control the device, which monitors the temperature remotely. The person must come and change the settings of this physical device by his hand; it can’t be controlled by a program usually.

Therefore, the idea of this project is the ongoing monitoring of the temperature in the Supercomputer Center and sending a warning to the engineer if it is necessary. The message will inform about the high temperature, which allows offering the necessary activities to deal with the rise of the temperature. We must design the application, which sends a message to the engineer's e-mail address and mobile phone.

Research goal and objectives

The goal of the research is the development of the application for monitoring the temperature at the supercomputer center of South Ural State University.

For the reaching this goal we must solve the following objectives:

1) to describe the problem statement of temperature monitoring;

2) to study the modern tools of application developmentfor Arduino microcontroller;

3) to design circuit diagram, deployment diagram and use case diagram for the application for monitoring the temperature;

4) to develop the structure of database for keeping history of temperature changes and current temperature;

5) to develop the interface of the client and administrator applications;

6) to implement and test the application.

The practical significance

This project is useful for all rooms that need the certain temperature. This project can be useful because it contains the possibility to remote threshold temperature from any computer connected to the server and the possibility to send email and SMS to the mobile phones.

This system can be used to monitor temperature in the supercomputer room, which contains many servers, which needs to work in environment with a certain temperature in order to protect it from damage. Also, you can take an advantage of this system to monitor the temperature in the intensive care rooms in hospitals, animal husbandry to control the degree of private rooms such as poultry temperature, in the process of storage of food items that require a certain temperature to keep the suitability of products and so on.

The application can be established at any amount of computers connected with the server.

Very useful possibility is that the user can change the threshold temperature through the application.

Structure of the thesis

The thesis consists of four chapters, introduction, conclusion andreference list.In the first chapter,the problem statement is represented,as well as the description of Arduino microcontroller and configuring of Arduino project window.

In chapter two, there is a description of functional requirements, circuit diagram, use case diagram, deployment diagram,and scheme of databaseand development of the interface of client and administrator applications.

In chapter three, we show preparation for the implementation, the program for microcontroller, the program for sending SMS and e-mails, and queries for the database.

Chapter Four is devoted to the testing of the application, the used methods of testing, and screen forms of the application.

The thesis has 38pages; the list of references contains 20 resources.

1. The analysis of the subject area

1.1. The problem statement

The range of temperature must be stable to prevent damage to the data center servers. Any electronic device transitioning from a cold temperature to a warm temperature could form water condensation inside the servers endangering the data center equipment to short out. While cold temperatures do not adversely affect electronic hardware, extremely high temperatures will cause overheating and, in event, failure of the electronic device[15].

In the Supercomputer Center, we need a continuous monitoring system for temperature to protect the servers in this room from damage due to high temperature, especially in the summer.

Therefore, the idea of this project is the ongoing monitoring of the temperature in the Supercomputer Center and sending a warning to the engineer if it is necessary. The message will inform about the high temperature, which allows offering the necessary activities to deal with the rise of the temperature. We must design the application, which sends a message to the engineer's e-mail address and mobile phone.

There are many systems, which monitor the temperature in rooms in the market, but none of them can inform the engineers and other staff about problems in their absence.

These systems can produce the alarming sound; can also automatically switch on/off the coolers. Simultaneously, they can’t send e-mails or SMS. In addition, it is impossible to control the device, which monitors the temperature remotely. The person must come and change the settings of this physical device by his hand; it can’t be controlled by a program usually.

My goal isto develop and manufacture the physical system for temperature controlling remotely, by the special application. In order to do that, I have to develop the circuit, to choose the microcontroller, to study how to create programs for it, the study how to create client applications for controlling the real physical device in a real-time mode and exchange the data between computers and device.

1.2.Description of Arduino microcontroller

The Arduino microcontroller [19] is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free [2].

The Arduino project was started in Italy to develop low cost hardware for interaction design. The Arduino home page contains information about the Arduino hardware of several types [11]. Fig.1 shows the Arduino microcontroller of type I, used in my project.

Fig. 1.Arduino board

The Arduino Nano ATmega328 microcontroller operates at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM(Electrically Erasable Programmable Read-Only Memory). EEPROM is a type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed.

Specifications of Arduino Nano ATmega328 microcontroller[8]:

1) ATmega328 type of microcontroller;
2) operating voltage 5v;
3) digital I/O pins 14;
4) analog input pins 8;
5) DC current per I/O 40 mA;
6) flash memory 32 KB;
7) clock speed 16 MHz;

8) EEPROM 1KB;
9) length 45 mm;
10) width 18 mm;
11) weight 5 g.

More information about Nano type of Arduino microcontroller is represented at [6].

The Arduino programming language is a simplified version of C/C++. The programmer, who knows C, can easily make programs for Arduino[17].

An important feature of the Arduino is that the programmer can create a control program on the host PC, download it to the Arduino device, and then this program will run automatically. The last program the programmer saved in the microcontroller will run. The programmer connects the Arduino board to the host PC in order to develop and debug his program by USB cable.

1.3. Configuring of Arduino project window

This is circuit diagram to my project which usedArduino [18]board Nano ATmega 328 connected to the host PC in order to develop and debug his program by USB cable. Fig. 2 shows the circuit diagram.

Connection of the Arduino board to the computer goes through the following steps.

Firstly, it is necessary to configure the computer to recognize the Arduino board. If the programmer want to install the new driver, he skips this step. The computer will not recognize the Arduino, and the programmer will not be able to upload the code to the Arduino microcontroller.

The steps of connection are the following:

1. Download and install Arduino software from the page with software available for downloading [4]. The programmer must download the file and extract the archive on the C drive.

2. Connect the board to the computer. PWR LED should light up.

3. Wait until the operating system Windows install driver for Arduino board.

-

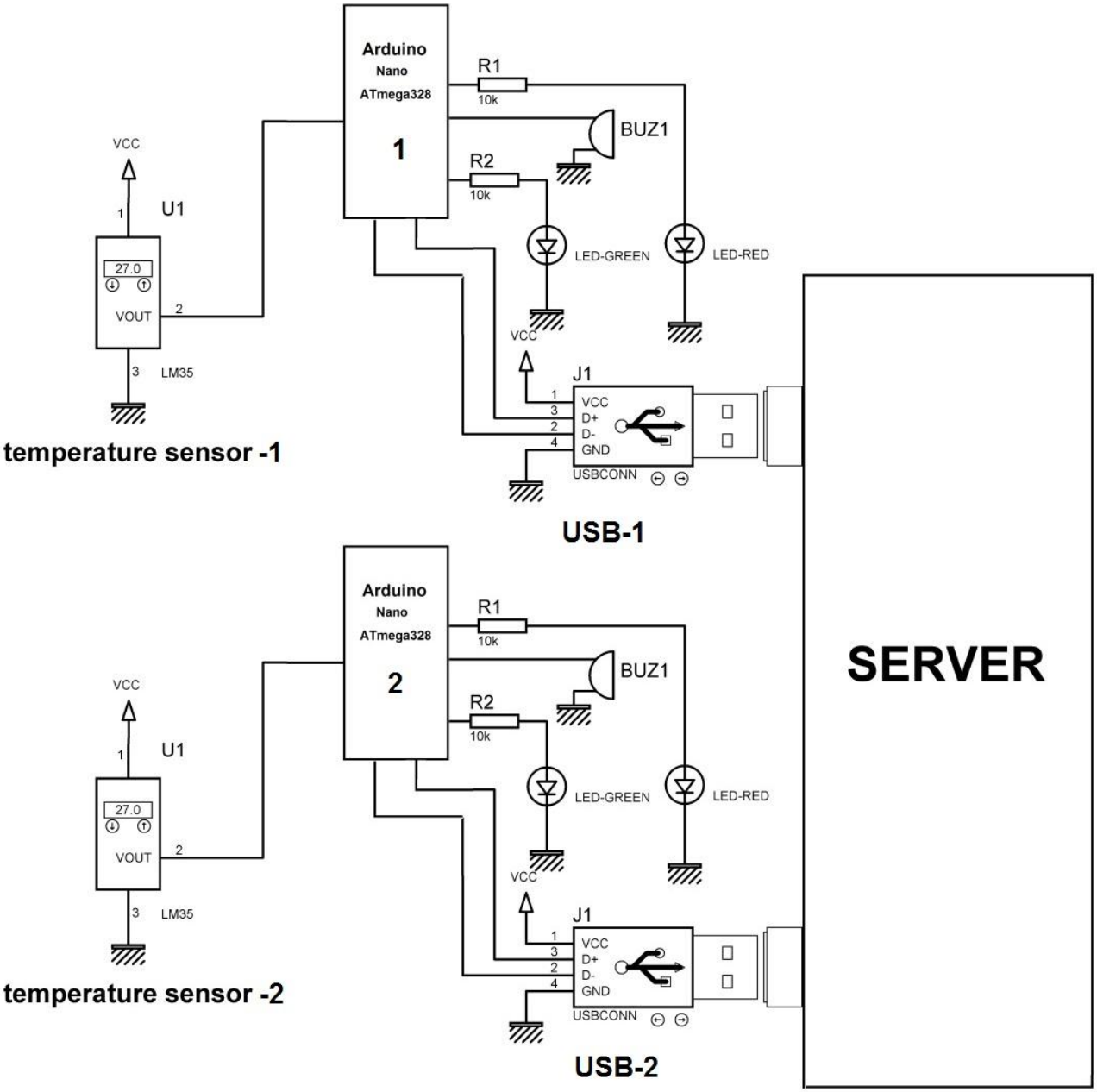


Fig. 2. Circuit diagram

After installing the driver, there must be following steps.

1. Open the program Arduino which was installed. Select “Tools”, after that choose “Board” from the options and select the type of Arduino board from the list. Fig. 3 shows the type of Arduino that used by me.

2. Choose the processor type from the menu “Tools” – “Processor” (fig. 4). I chose the type ATmega328.

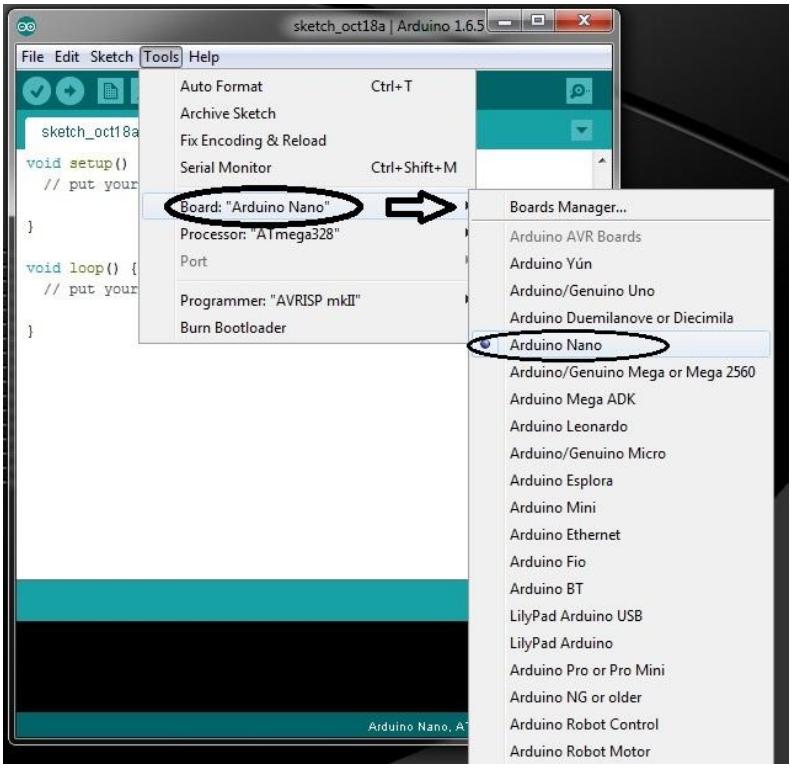


Fig. 3. Chose Arduino type

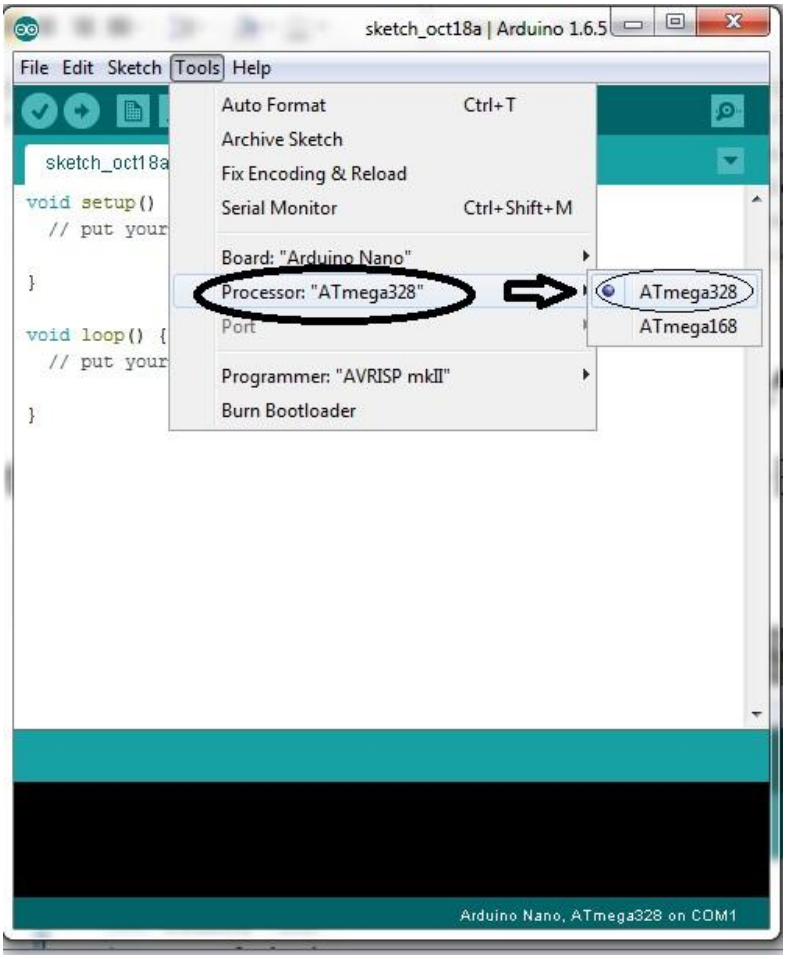


Fig. 4. Selection of the processor

3. Choose the port name from the menu “Tools” – “Port” (fig. 5). I chose the port COM3, it is a number of serial port used by computer.

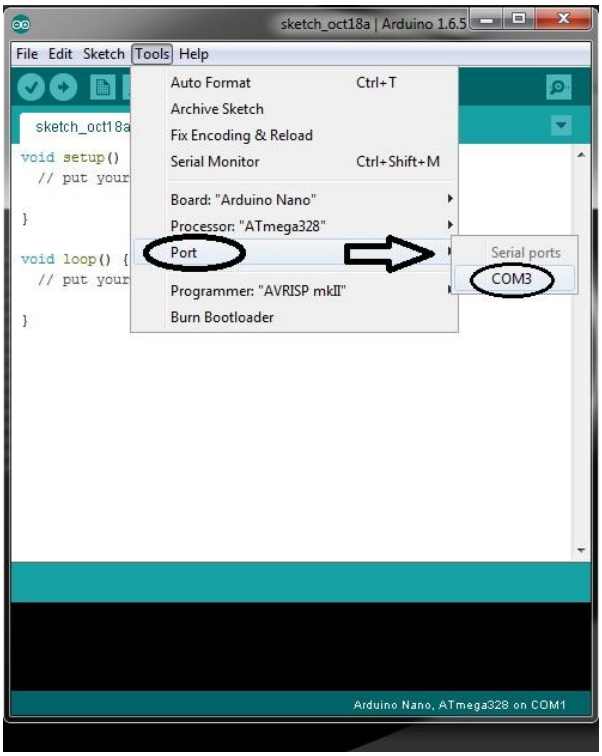


Fig. 5. Choosing the serial port number

After that, the programmer must write the program codes inside the project window as shown in fig. 6.

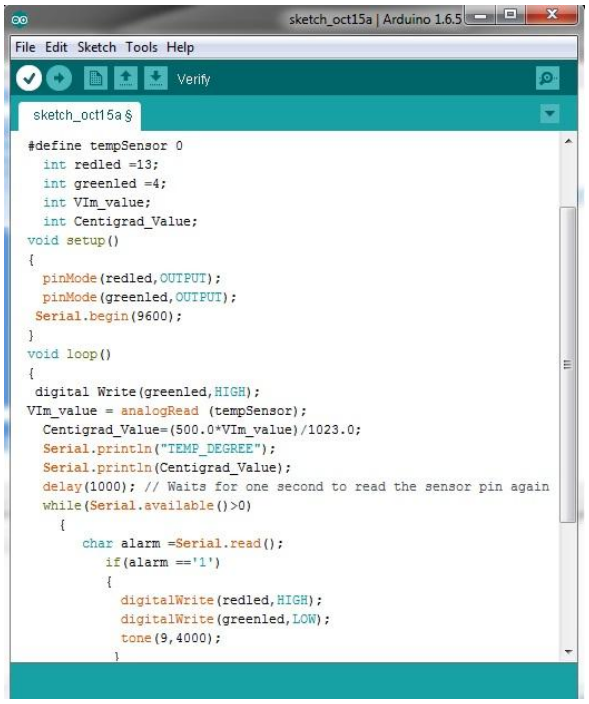


Fig. 6.The program codes

After the program is ready, the programmer must verify it, if it has errors, whether it is correct or not as shown in fig. 7.

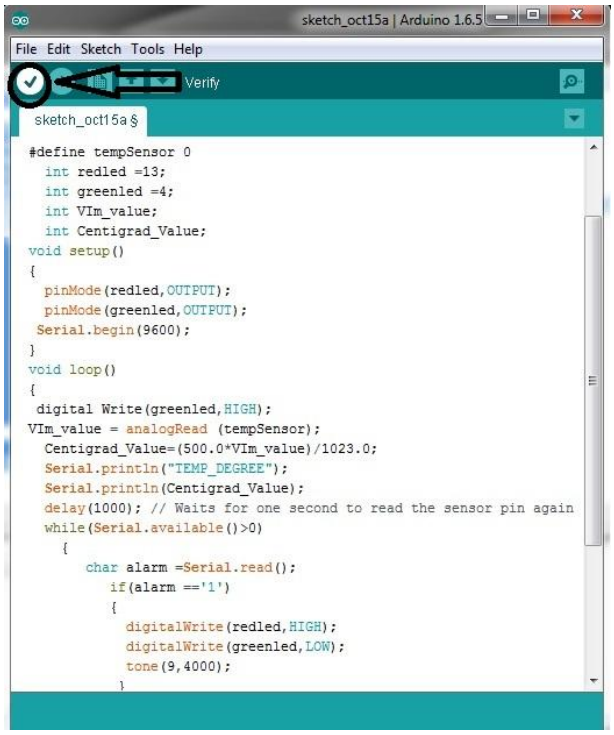


Fig. 7. Verifying the program

When the program is already verified, the programmer can upload it to the Arduino microcontroller as it is shown in fig. 8.

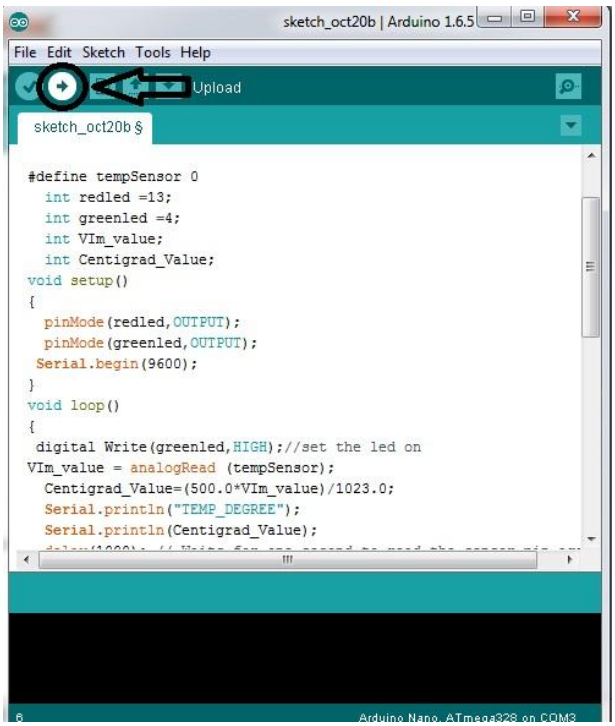


Fig. 8. Upload the program

2. Development of application

2.1.Functional requirements

There must be two applications: for the administrator and the engineer.

The application for administrator contains the following basic functions:

1) show the current temperature from sensor 1;
2) show the current temperature from sensor 2;
3) show the threshold temperature;
4) set a new threshold temperature;
5) show the information from the table “History”;
6) save new value of the threshold in the tables “History” and “Client_Changes”.

The application for client contains the following basic functions:

1) show the current temperature from sensor 1;
2) show the current temperature from sensor 2;
3) show the threshold temperature;
4) set a new threshold temperature;
5) save new value of the threshold in the tables “History” and “Client_Changes”.

2.2. Circuit diagram

In my project, I used two groups of components, to each groupIshall take the temperature via thetemperature sensor LM35. It will connect with theArdui-nomicrocontroller [16], and I shall use the USB-port to connect with the server where my application will work. The project circuit is represented in fig. 9.

There are 13 points at the picture:

1) Pin 13 connected with red led through resistance 10 kΩ;
2) Pin A0 received analog signal from temperature sensor;
3) Pin 5v as power source to temperature sensor;
4) Pin 4 connected with green led through resistance 10 kΩ;
5) Microcontroller ATmega 328;
6) USB-port;

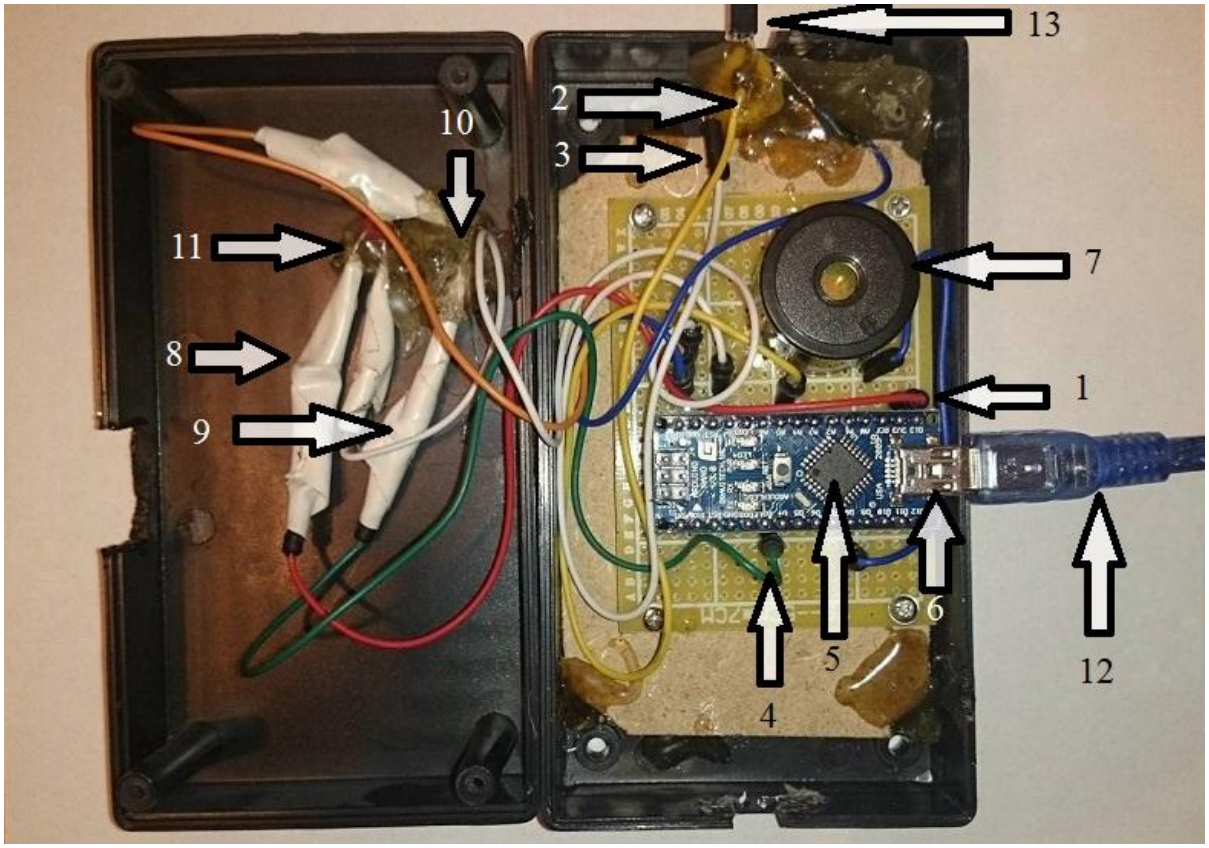


Fig. 9. The project circuit

7) Buzzer connected with pin 9;
8) Resistance 10 kΩ connected between pin 13 and red led;
9) Resistance 10 kΩ connected between pin 4 and green led;
10) Green led;
11) Red led;
12) USB cable;
13) Temperature sensor LM35.

The temperature sensor LM35 is represented in fig 10. It is a sensor to measure the temperature of the air around it.

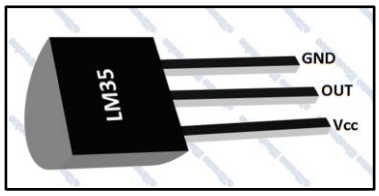


Fig. 10. Temperature sensor

The Microcontroller ATmega328inside Arduinoboard converts the input analog signal from the sensor to the digital signal, so that I can use it in the computer, according to mathematical equations, which are inside the microcontroller. The microcontroller ATmega328 is shown in fig. 11.

Fig. 11.Arduino board

USB-Port is used to connect the Arduino board and the server computer to send output data from microcontroller to the server application.

Thefinalshape of thedeviceused inthe project is represented in fig. 12.

Fig. 12.The project device

2.3. Use case diagram

We developed use case diagram [1] for application,the use case diagram for application shows that there are two actors “Administrator” and “User”.The administrator can see “Current Temperature”,“Current threshold Temperature”,“Set new threshold temperature” and “See history”. The User can see “Current Temperature”,“Current threshold Temperature” and“Set new threshold temperature”. Use case diagramfor using the application is represented in the (fig. 13)

The “User” can see all functions which administrator see it except the function “See history”.

The relation between the “Current threshold Temperature” and“Set new threshold temperature” it is extend.

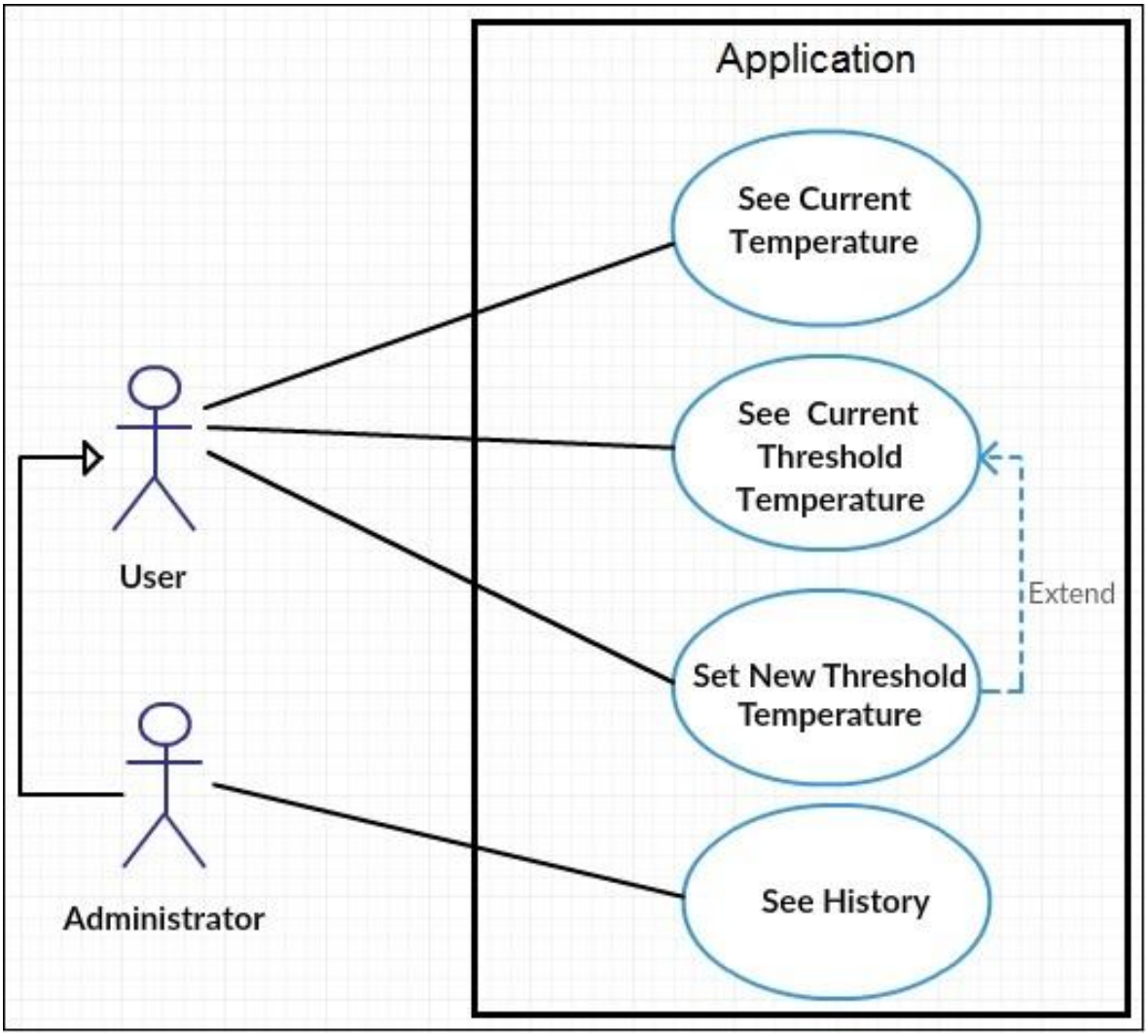


Fig. 13. Use case diagram for application

2.4. Deployment diagram

The deployment diagram shows the sensor, which measures the temperature and sends it to the microcontroller. Microcontroller sends data after converting from analog to digital signal to the server. The server will send a message to the engineer's e-mail address and an SMS to his mobile phone. These devices are available through clouds (fig. 14).

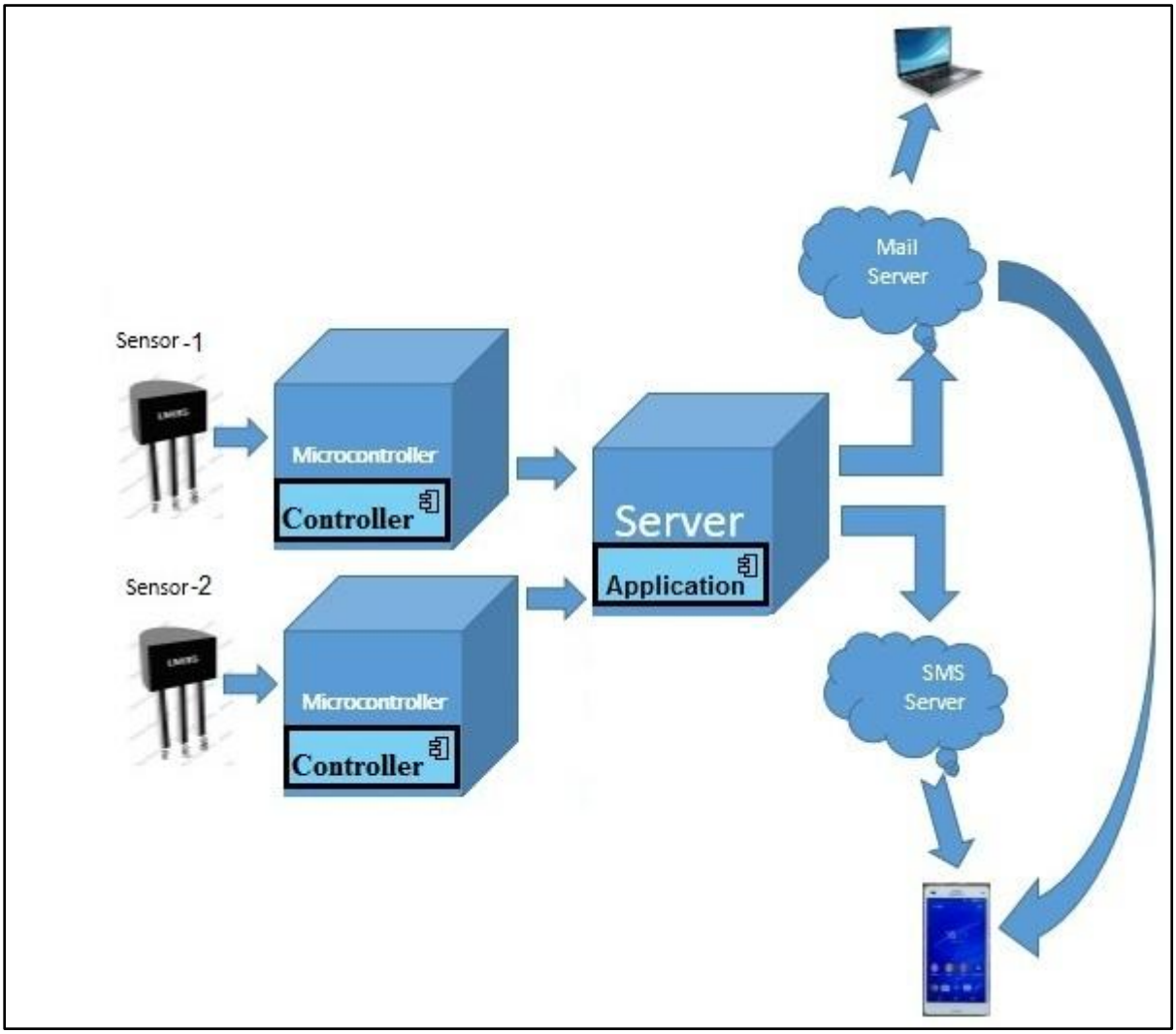


Fig. 14. Deployment diagram

2.5. Development of the database

For keeping the history about the system's work and user's changes I developed the database with two tables: "History" and "Client_changs".

The table "History" contains information about the data that received fromthe serial port, whichis connected with Arduino board. It contains four columns:

1) id;
2) temperature;
3) mdate;
4) threshold;
5) sensor.

The table “History” is represented in fig. 15.

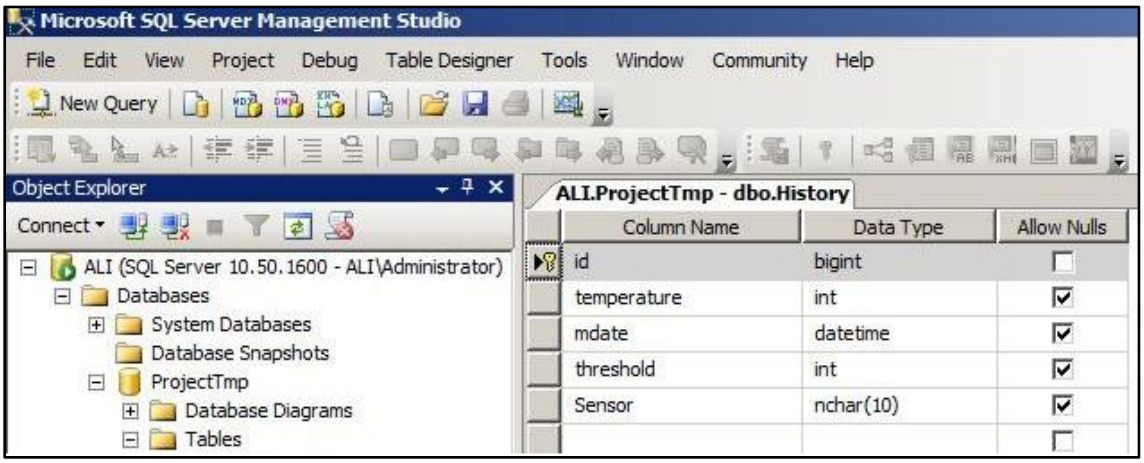


Fig. 15. Structure of the table “History”

The table “History” saves the current temperature and the time to this read and threshold value, that represented infig. 16.

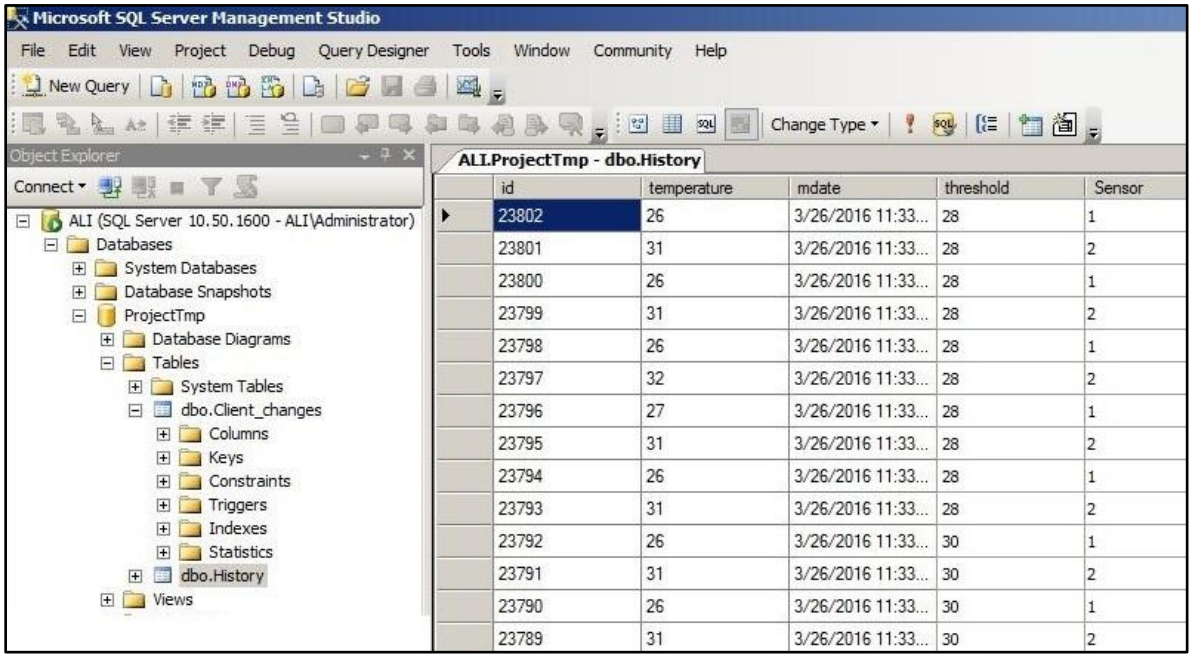


Fig. 16.Contents of the table “History”

The table “Client_changes” contains information about the changes of the threshold by a client or administrator.

It contains twocolumns:

1) id;
2) threshold.

The table “Client_changes” is represented in fig. 17.

The table “Client_changes” saves the threshold changes by administrator and client (fig. 18).

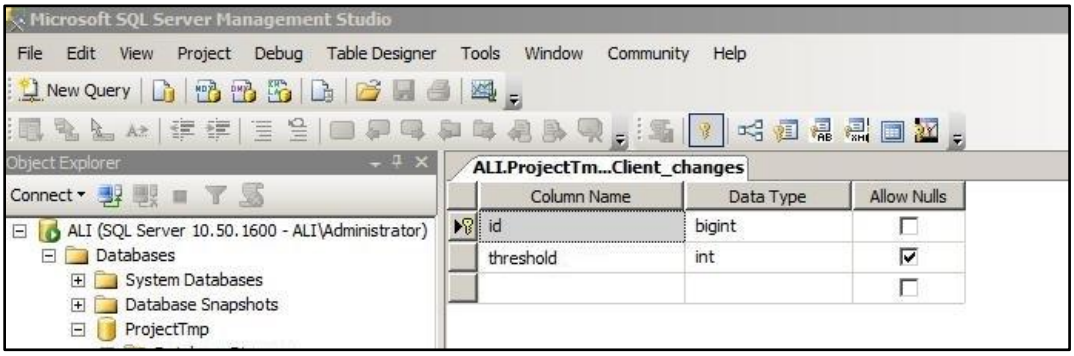


Fig. 17. Structure of the table “Client_changes”

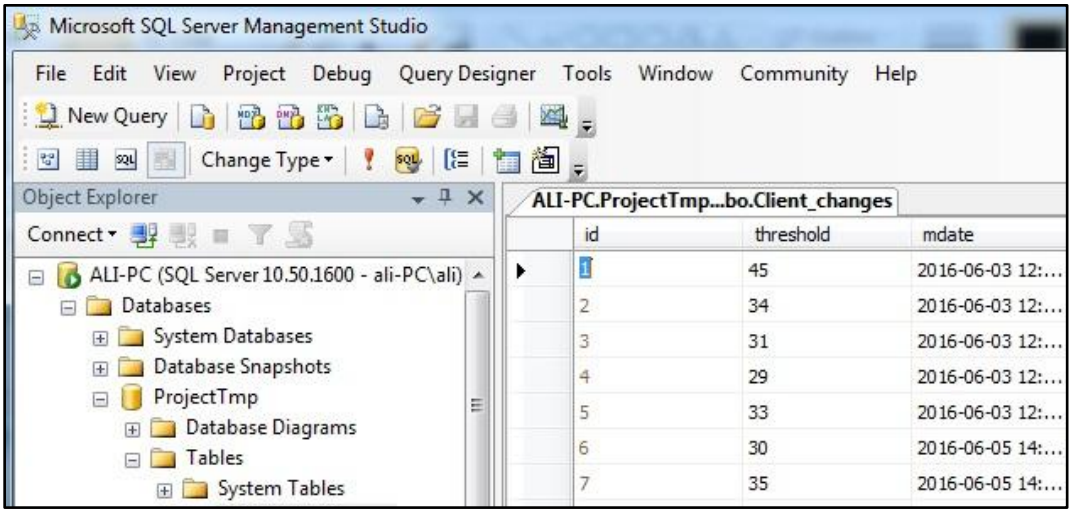


Fig. 18.Contents of the table “Client_changes”

The item value “threshold” in the table “Client_changes” it is will be the equel the item value “threshold” in the table “History” that's mean the application for administrator and application for client there are see the “threshold” from the table “Client_changes” and the relation between item “threshold in two table it is one to many the fig. 19.”

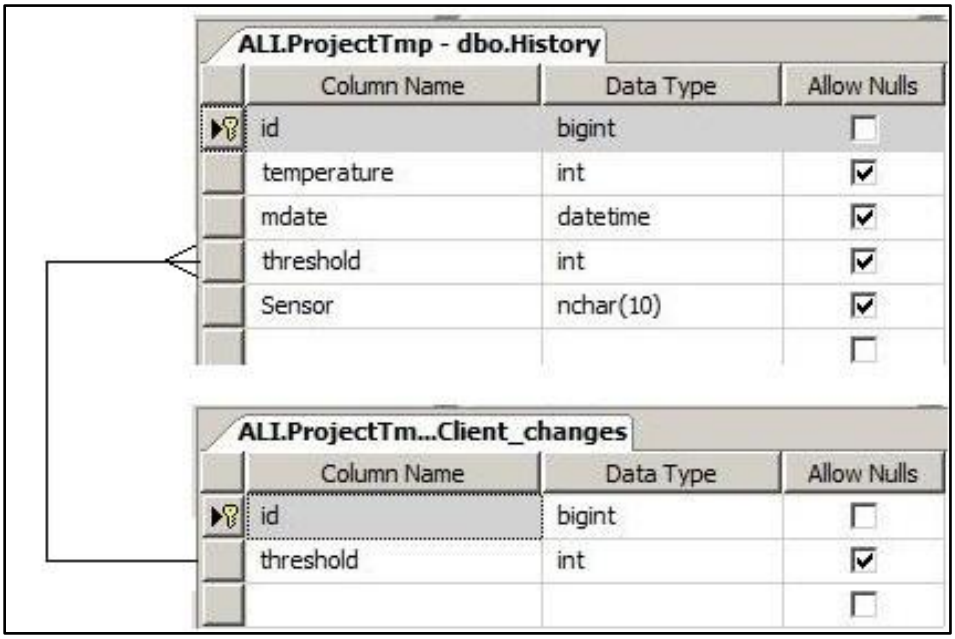


Fig. 19. Relation between “threshold”in the tables “History” and“Client_changes”

2.6. Development of the interface of client and administrator applications

I shall implement future viewsof an application.

Fig. 20 shows the administrator application window for temperature monitoring system.

Fig. 21 shows the window “History” for temperature monitoring system. It shows information about the temperature in previous time and updates which were made by administrator (button “Refresh”).

Fig. 22 shows the client application window for the temperature monitoring system.

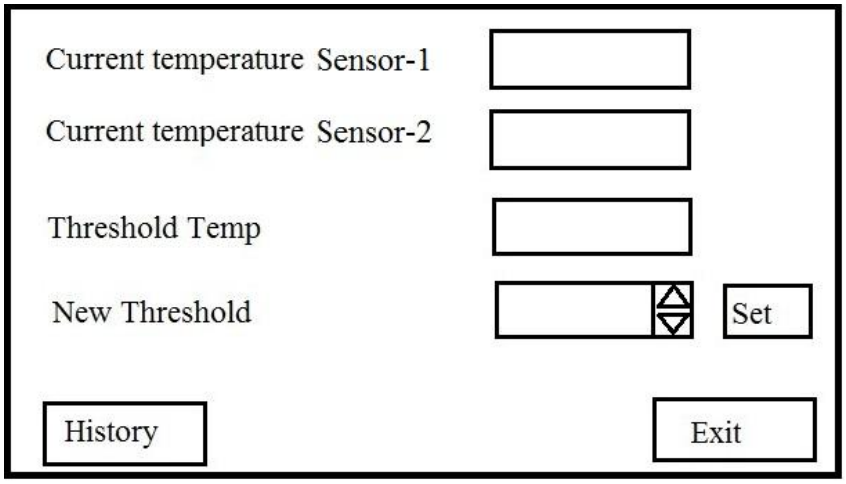


Fig. 20. The administrator window

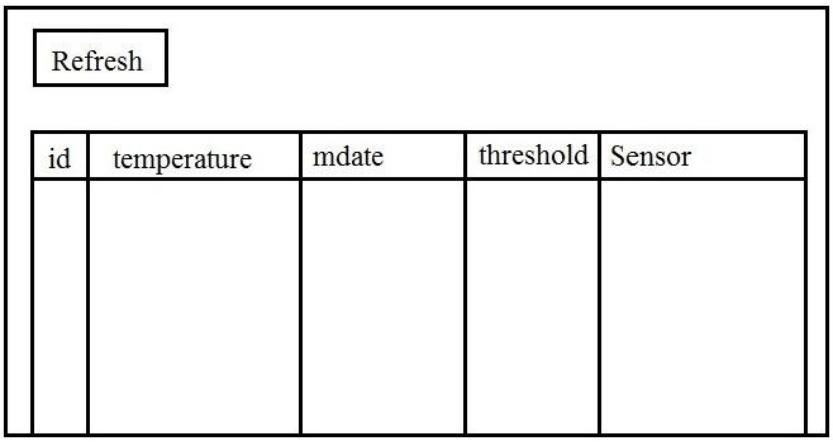


Fig. 21. The window “History”

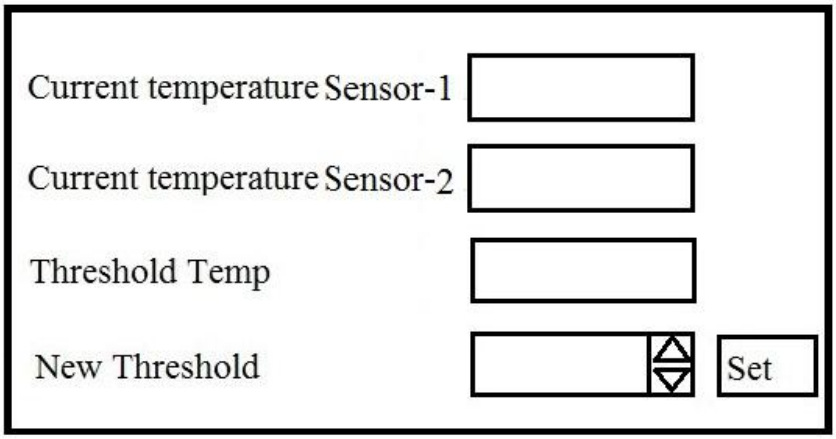


Fig. 22. The client window

3. Implementation of applications

3.1. Preparation for the implementation

This project needs the special environment and need the device containing the temperature sensor and microcontroller to measure the temperature degree, which, the application will monitor it in my project.

I used the operating system Windows Server 2008 [14]. Server will exchange data with administrator's and client's computers by IPof these computers. I save and access to all information about temperature in the project by using Structured Query in database [5]. I built the database on this operation system using Microsoft SQL 2008 [12].

The administrator's application works at the administrator computer, whichconnects with the project devices. The client application can run at another computer connected with server without necessity of connecting with the project devices.

I used the Visual Studio.NET [13] to write my applications program in Visual Basic language [7].

3.2. Program for microcontroller

I used Arduinoboardthat containmicrocontroller type ATmega 328[10]. Arduino programs can be divided in three main parts:

1) Structure (setup() and loops());
2) Values (variables and constants);
3) Functions.

More information about these parts is represented at the website [9].

Program codes which I have written are represented in fig. 23.

In this program, there are the following commands.

`#define tempSensor 0` – defines the initial value for the temperature sensor, it is zero.

```
/*This projects to monitoring the temperature
of the super computer center using v.b.net and arduino
under the supervisor Ivanova Olga Nikolaevna designed by  Ali Mohammed Mozan 12/6/2015*/

 #define tempSensor 0 //define the initial value from tep sensor it is zero
   int redled =13;
   int greenled =4;
   int VIm_value;
   int Centigrad_Value;
 void setup()
 {
   pinMode(redled,OUTPUT);
   pinMode(greenled,OUTPUT);
  Serial.begin(9600);
 }
 void loop()
 {
  digital Write(greenled,HIGH);//set the led on
 VIm_value = analogRead (tempSensor);
   Centigrad_Value=(500.0*VIm_value)/1023.0;
   Serial.println("TEMP_DEGREE");
   Serial.println(Centigrad_Value);
   delay(1000); // Waits for one second to read the sensor pin again
   while(Serial.available()>0)
     {
        char alarm =Serial.read();
           if(alarm =='1')
           {
             digitalWrite(redled,HIGH);//set the led on
             digitalWrite(greenled,LOW);//set the led off
             tone(9,4000);
            }
           else if(alarm=='0')
           {
                 digitalWrite(redled,LOW);
                 digitalWrite(greenled,HIGH);
                 noTone(9);

           }
     }
 }
```

Fig. 23. Program for microcontroller

`intredled =13` – the red led is connected to the pin 13.

`intgreenled =4` – the green led is connected to the pin 4.

`intVIm_value` – the value of the analog read voltage from the temperature sensor.

`intCentigrad_Value` – the value of a degree of Celsius temperature. The Celsius and Centigrad values are given in [3].

`pinMode(redled,OUTPUT)` – sets the pin which the red led connected with it as output.

`pinMode(greenled,OUTPUT)` – sets the pin which the green led connected with it as output.

`Serial.begin(9600)` – sets the data rate in bits per second (baud) for serial data transmission. For communicating with the computer I use the rate 9600.

`Digital Write(greenled, HIGH)` – sets the led on.

`VIm_value = analogRead (tempSensor)` – sets the analog read voltage `VIm_value` from initial value which is equal zero.

`Centigrad_Value=(500.0*VIm_value)/1023.0` – calculation of Centigrad_Value according to this equation. 5v = 500mv (5V is the highest value the temperature sensor gives as voltage, this value will multiply with the analog read voltage (VIm_value) and divided by 1023 to become a digital value.

`Serial.println("TEMP_DEGREE")`– prints to the serial port the sentence “TEMP_DEGREE”.

`Serial.println(Centigrad_Value)` – prints to the serial port the sentence “Centigrad_Value”.

`delay(1000)` – waits for one second to read the sensor pin again.

`char alarm = Serial.read()` – reads the char as alarm name from the serial port.

`digitalWrite(redled,HIGH)` – sets the red led on.

`digitalWrite(greenled,LOW)` – set the green led off.

`tone(9,4000)` – runs the buzzer on pin 9 in frequency 4000Hz.

`digitalWrite(redled,LOW)` – set the red led off.

`digitalWrite(greenled,HIGH)` – set the green led on.

`noTone(9)` – turns off the buzzer.

3.3. Program for sending SMS and e-mails

Firstly, I decided to develop the application that sends e-mail and SMS to the engineer's mobile. The application is written in Visual Basic [20]. The codes of the program is shown in fig. 24.

VB.Net allows sending e-mails from the application. The System.Net.Mail namespace contains classes used for sending e-mails to a Simple Mail Transfer Protocol (SMTP) server for delivery. The SmtpClient class allows applications to send e-mail by using the Simple Mail Transfer Protocol (SMTP).

MailMessage represents an e-mail message that can be sent using the SmtpClient class.

MailAddress represents the address of an electronic mail sender or recipient.

SmtpClient allows applications to send e-mail by using the Simple Mail Transfer Protocol (SMTP).

EnableSsl specifies whether the SmtpClient uses Secure Sockets Layer (SSL) to encrypt the connection.

Credentials gets or sets the credentials used to authenticate the sender.

Send(MailMessage) sends the specified message to an SMTP server for delivery.

SmtpException represents the exception that is thrown when the SmtpClient is not able to complete a Send or SendAsync operation.

```
Imports System.Net.Mail
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim EmailMessage As New MailMessage()
        Try
            EmailMessage.From = New MailAddress("alimozan44@gmail.com")
            EmailMessage.To.Add("ali_mozan7819@yahoo.com,79058301021@sms.beemail.ru,alimozan44@gmail.com")
            EmailMessage.Subject = "The subject"
            EmailMessage.Body = "The Temprutare In The Super Computer Room It Is High "
            Dim SMTP As New SmtpClient("smtp.gmail.com")
            SMTP.Port = 587
            SMTP.EnableSsl = True
            SMTP.Credentials = New System.Net.NetworkCredential("susu57772@gmail.com", "pasword ")
            SMTP.Send(EmailMessage)
        Catch ex As Exception

        End Try

    End Sub

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub
End Class
```

Fig. 24. Code of the application for sending e-mail and SMS

3.4. Queries for the database

This is the query to select all items from table “History” when the administrator clicks the button “History” at his main form (fig. 25).

```
strSelect = "SELECT * FROM History"
```

Fig.25. Query “History”

This is the query to select all items from table “Client_changes”, which contains the field “threshold” (new threshold temperature) (fig. 26). This query runs when the client or administrator change threshold.

```
Private Function loadThreshold() As Integer
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim ds As DataSet
    Dim da As New SqlDataAdapter
    Dim dt As New DataTable
    Dim thresh As Integer
    con = Conn()
    con.Open()
    cmd = con.CreateCommand
    cmd.CommandText = "SELECT * FROM Client_changes Order by id desc"
    da.SelectCommand = cmd

    ds = New DataSet()
    ds.Tables.Add(dt)
    da.Fill(dt)
    ' MsgBox(lgname)

    da.Fill(ds, "Client_changes")

    If dt.Rows.Count <= 0 Then
        Return 0
    Else

       thresh = ds.Tables("Client_changes").Rows(0).Item("threshold")

        Return thresh
    End If
    con.Close()

End Function
```

Fig. 26. Query “Client_changes”

4. Testing of application

4.1. The used methods of testing

There are two applications: for the administrator and the client.

Functional system tests should be based on the coverage of the functionality described in the requirements. Each test of my system contains input and output information.Therefore, I compare the actual results and the expected resultsto administrator application (tab. 1) and to client application (tab. 2).

Table 1.The protocol of functional testing of the administrator application

No.	Function	Expected result	Obtained result	Conclusion
1.	Show the current temperature sensor-1	The administrator applicationdisplays the current temperature in the“Current temperature” textbox	The administrator application displaysthe current temperature in the “Current temperature” textbox	The function works
2.	Show the current temperature sensor-2	The administrator application displays the current temperature in the “Current temperature” textbox	The administrator application displays the current temperature in the “Current temperature” textbox	The function works
3.	Show the threshold temperature	The administrator application displays the last saved value of the threshold temperature in the “threshold” textbox	The administrator application displays the last saved value of the threshold temperature in the “threshold” textbox	The function works
4.	Set a new threshold temperature	The administrator application save the new value of threshold temperature in the tables “History” and “Client_changes” and display it in “threshold” textbox	The administrator application save the new value of threshold temperature in the tables “History” and “Client_changes” and display it in “threshold” textbox	The function works

5.	Show the information from the table “History”	The administrator application opens the new form with information of the table “History”	The administrator application opens the new form with information of the table “History”	The function works
6.	Save new value of the threshold in the tables “History” and “Client_Changes”.	Administrator application displays the new value in the “threshold” textbox	Administrator application displays the new value in the “threshold” textbox	The function works

Table 2. The protocol of functional testing of the client

No.	Function	Expected result	Obtained result	Conclusion
1.	Show the current temperature sensor-1	The client application displays the current temperature in the “current temperature” textbox	The client application displays the current temperature in the “current temperature” textbox	The function works
2.	Show the current temperature sensor-2	The client application displays the current temperature in the “current temperature” textbox	The client application displays the current temperature in the “current temperature” textbox	The function works
3.	Show the threshold temperature	The client application displays the last saved value of the threshold temperature in the “threshold” textbox	The client application displays the last saved value of the threshold temperature in the “threshold” textbox	The function works
4.	Set a new threshold temperature	The client application save the new value of threshold temperature in the tables “History” and “Client_changes” and display it in “threshold” textbox	The client application save the new value of threshold temperature in the tables “History” and “Client_changes” and display it in “threshold” textbox	The function works

5.	Save new value of the threshold in the tables “History” and “Client_Changes”.	Client application displays the new value in the “threshold” textbox	Client application displays the new value in the “threshold” textbox	The function works

4.2. Screen forms of the application

The window application for administrator is represented in fig. 27.

When the administrator clicks the button “History”, he sees the special form (fig. 28).

The window application for client is representedin fig. 29.

On default the background color of the textbox “Current temperature” is green. It means that the current temperature is normal. When the temperature is more than or equal to the threshold value then the background of the textbox with the temperature will change its color from green to red (shown in fig. 30).

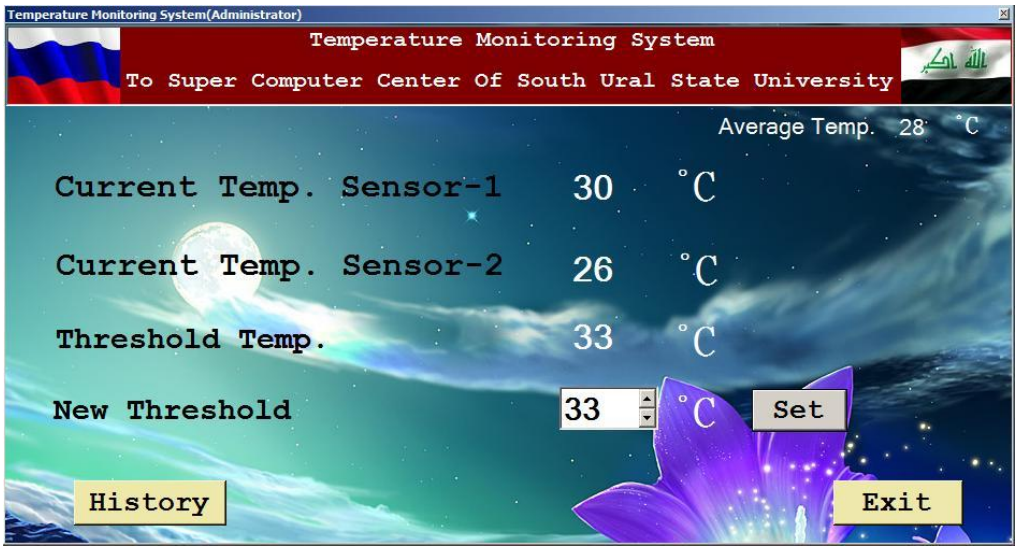
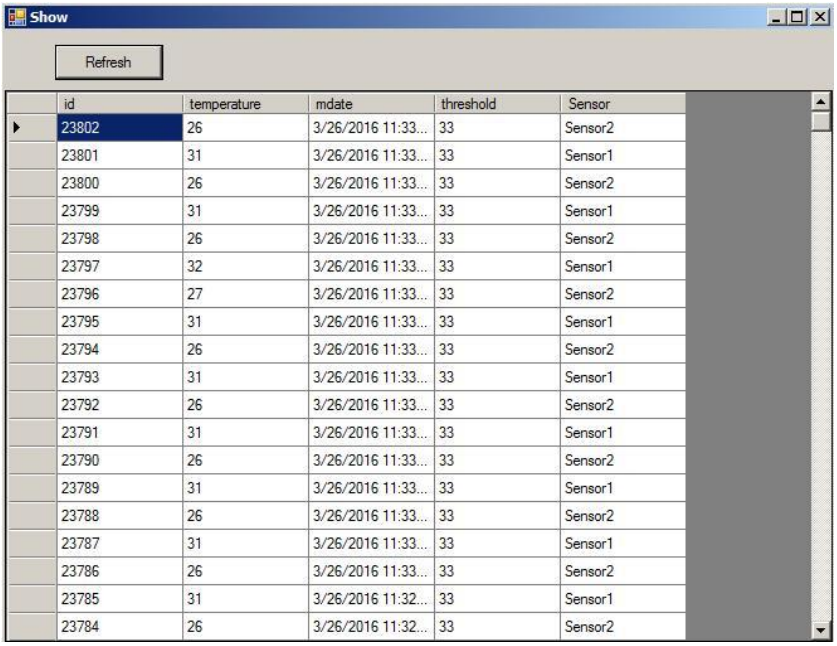


Fig. 27.Application for administrator



id	temperature	mdate	threshold	Sensor
23802	26	3/26/2016 11:33...	33	Sensor2
23801	31	3/26/2016 11:33...	33	Sensor1
23800	26	3/26/2016 11:33...	33	Sensor2
23799	31	3/26/2016 11:33...	33	Sensor1
23798	26	3/26/2016 11:33...	33	Sensor2
23797	32	3/26/2016 11:33...	33	Sensor1
23796	27	3/26/2016 11:33...	33	Sensor2
23795	31	3/26/2016 11:33...	33	Sensor1
23794	26	3/26/2016 11:33...	33	Sensor2
23793	31	3/26/2016 11:33...	33	Sensor1
23792	26	3/26/2016 11:33...	33	Sensor2
23791	31	3/26/2016 11:33...	33	Sensor1
23790	26	3/26/2016 11:33...	33	Sensor2
23789	31	3/26/2016 11:33...	33	Sensor1
23788	26	3/26/2016 11:33...	33	Sensor2
23787	31	3/26/2016 11:33...	33	Sensor1
23786	26	3/26/2016 11:33...	33	Sensor2
23785	31	3/26/2016 11:32...	33	Sensor1
23784	26	3/26/2016 11:32...	33	Sensor2

Fig. 28. The “History” window

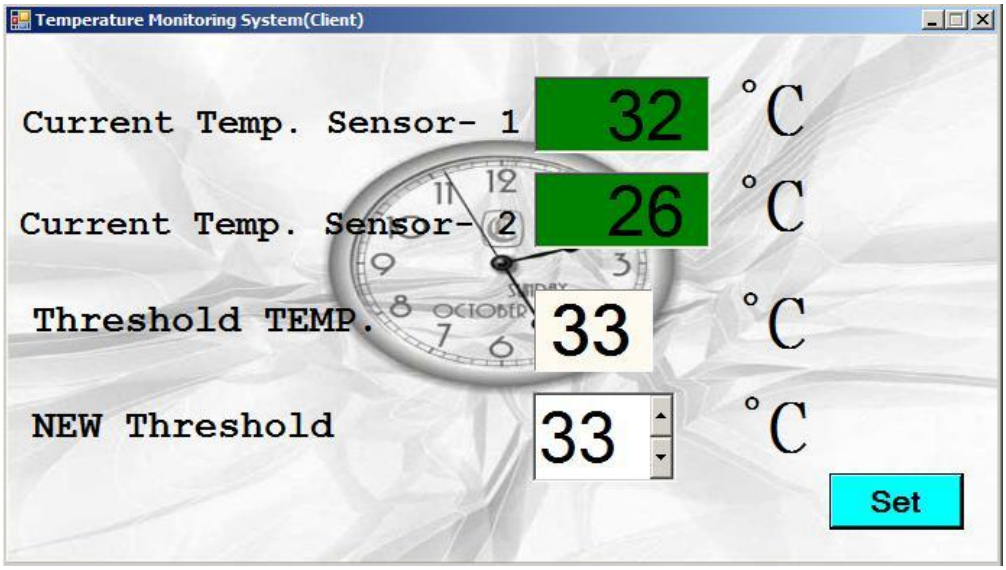


Fig. 29. Application for client

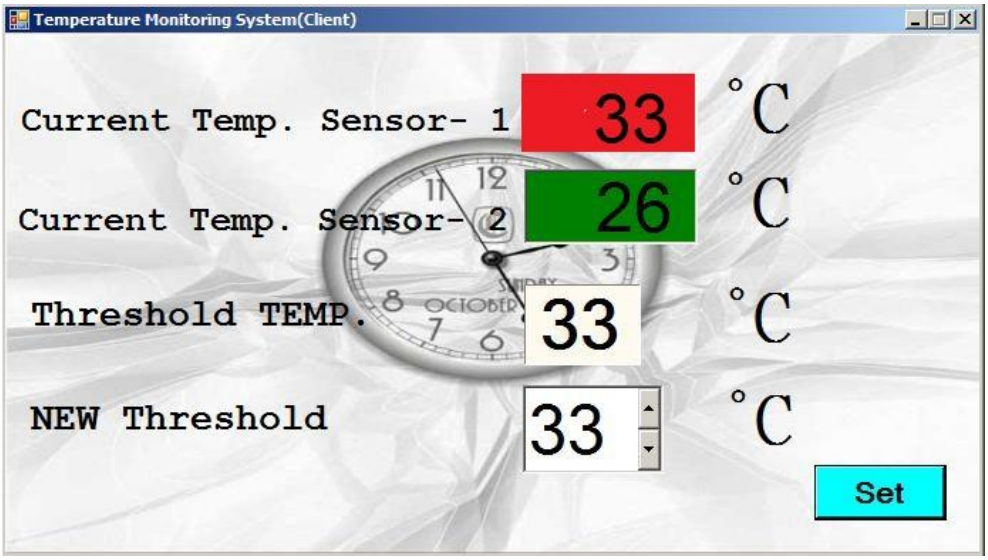


Fig. 30. The color change to the temperature text

The administrator application has a special icon located on the desktop (fig. 31).

Fig. 31. Icon of the administratorapplication

The client application has a special icon located on the desktop (fig. 32).

Fig. 32. Icon of the client application

Conclusion

In this thesis, the application for monitoring the temperature at the Supercomputer center of South Ural State University was proposed to design and setup automatic alarming system.

Most probablyin summer days,the supercomputer center requires a monitoring system, which monitors the rise of high temperatures in the server rooms.

There will not be alarmsof the responsible engineers' staff in their absence. The existing analogues can play the most important role of alarming by the sound system and mechanically switch on/off the cooler system. At the same time, they cannot send e-mails or SMS to the engineers. On the other hand, it is impossible to control the device, which observed the temperature, comfortably and easy. The person must come and change the settings of this physical device by himself; a program could not control it regularly.

So that, the aim of this project is to develop an applications, preventing the danger of the high temperaturein the supercomputer center and alarming the responsible workers by sending email or massage to the workers' phones.

The goal of the research is the development of the application for monitoring the temperature at the supercomputer center of South Ural State University.

For the reaching this goal we solved the following objectives:

1) described the problem statement of temperature monitoring;

2) studied the modern tools of application development for Arduino microcontroller;

3) designed circuit diagram, deployment diagram and use case diagram for the application for monitoring the temperature;

4) developed the structure of databasefor keeping history of temperature changes and current temperature;

5) developed the interface of the client and administrator applications;

6) implemented and tested the application.

The perspectives for the application are the following:

1) to control the temperature degree by air condition devices through the application;

2) to make the mobile version for the application.

Reference list

1. Bennett S., Skelton J., Lunn K. UML. – USA, New York: McGraw-Hill, 2005. – 398 p.
2. Blum R. Arduino programming in 24 hours. – India: Indianapolis, Indiana Sams, 2015. – 419 p.
3. Difference between Celsius Centigrade. [Electronic resource] URL: http://chemistry.about.com/b/2014/01/29/difference-between-celsius-and-centigrade.htm(the date of access: 17.10.2015).
4. Download the Arduino Software. [Electronic resource] URL:https://www.arduino.cc/en/Main/Software (the date of access: 17.10.2015).
5. Garsia-Molina H., Ullman J.D., Widom J. Database systems. The complete book. 2nd Edition. – USA, New Jersey: Pearson Education Inc., 2009. – 1241 p.
6. General information about Arduino. [Electronic resource] URL: http://www.vertical-research.com/sites/default/files/General%20Information%20about%20Arduino_1.pdf (the date of access: 22.10.2015).
7. Halvorson M. Visual Basic 2010. – USA: Microsoft Press, 2010. – 529 p.
8. Informationabout specifications of Nano type of Arduino microcontroller. [Electronic resource] URL: https://www.arduino.cc/en/Main/ArduinoBoardNano (the date of access: 15.10.2015).
9. Language reference. [Electronic resource] URL: https://www.arduino.cc/en/Reference/HomePage (the date of access: 17.10.2015).
10. Morton J. The PIC Microcontroller. – UK: Newnes, 2005. – 223 p.
11. Official site of company Arduino. [Electronic resource] URL: http://www.arduino.cc/ (the date of access: 22.10.2015).

12. Official site of company Microsoft to download SQL Server 2008. [Electronic resource] URL: https://www.microsoft.com/en-us/download/details.aspx?id=30438 (the date of access: 22.10.2015).

13. Official site of company Microsoft to download Visual Studio.NET. [Electronic resource] URL: https://www.microsoft.com/en-us/download/details.aspx?id=23691/ (the date of access: 22.10.2015).

14. Official site of company Microsoft to download Windows Server 2008. [Electronic resource] URL: https://www.microsoft.com/en-us/download/details.aspx?id=5023 (the date of access: 22.10.2015).

15. Optimal Data Center / Server Room Temperature and Humidity. [Electronic resource] URL: http://www.ravica.com/blog/data-center/data-center-temperature-and-humidity-optimization/ (the date of access: 05.11.2015).

16. Oxer J., Blemings H. Practical Arduino. – USA: Apress, 2009. – 409 p.

17. ParabJ., Shinde S.,Shelake V., Kamat R., Naik G. Practical Aspects of Embedded System Design using Microcontrollers. – USA: Springer, 2008. – 147 p.

18. Purdum J. Beginning C for Arduino. – USA, New York: Springer Science + Business Media, 2012. – 262 p.

19. Wheat D. Arduino internals. – USA: t.i.a, 2011. – 372 p.

20. Zak D. Programming with Microsoft Visual Basic 2010. – USA: Cengage Learning, 2011. – 840 p.