Humidifier with a Web Interface

Marta Jóźwiak, David Testar Carbó, Iván Arrabal Camacho, Peter Gal

***Abstract*— Is it possible to make yourself a humidifier with a web interface for a server room? Does it requires special abilities or equipment? We will answer all those questions and more in this paper. We will go through, step by step, from choosing the type of humidifier, control, and web interface we are planning to build.

***Keywords*— humidifier, humidity, server room, web interface.

I. Introduction

THE aim of the project was to build a humidifier with a web interface for a data center of 80 m^3. It should maintain humidity between 40% and 70% (± 5). It also should be autonomous for two days and it should posses water and humidity alarms. Moreover, a good distribution of water vapour is expected. The humidifier should be compliant with the EU Directives 2006/42/CE 2006-05-17 and 2006/95/CE 2006-12-12. Humidifying Data Center is very important, because it is directly related to the security of IT equipment. If the relative humidity drops below 40%, the excess of static electricity may cause sparks which can damage servers and IT equipment. A high level of humidity is also inconvenient for electrical machines. Both situations may cause server downtime and may lead to serious equipment damage and, consequently to large cost.

II. Different types of humidifiers

There are several types of humidifiers available on market. We acquaint with all of them in order to choose one which would the most suitable for our purpose.

A. Evaporative Humidifiers

In evaporative humidifiers mist is produced by blowing water through a wick filter with the use of a fan. Hot air goes through a wet filter causing the increase of humidity and decrease of temperature. The mineral dust and bacteria get trapped on the filter before the water evaporates into the air. [1], [2], [3]

B. Impeller Humidifiers

In impeller humidifiers a fan blows water through a diffuser and, as a result, it produces fine droplets. [1], [2], [3]

C. Ultrasonic Humidifiers

Ultrasonic humidifiers use high-frequency vibrations to dissipate water into the air. This method is the quietest of all. [1], [2], [3]

D. Warm Mist Humidifiers

These humidifiers heat the water and expel steam into the room. Water is heated and mixed with the air as a vapour This method is very quiet and produces pure and clean mineral-free moisture output. [1], [2], [3]

E. Humidifiers with an Air Compressor

The water is blown into the air through nozzles that mix high-pressure air with water. Due to the high-pressure air, the water is dispersed of the water vapour trough the nozzle.

III. Controlling System

A. Programmable Logic Controller

PLC is a digital computer used for automation of electromechanical processes. [9] PLCs have standard inputs/outputs, and may include Ethernet connection. The most important feature of the PLC is their universal usability, but usually PLCs are used in industry. The main difference from other controllers is that PLCs are armored for severe conditions (such as dust, moisture, heat, cold) and have the facility for extensive input/output (I/O) arrangements allowing the PLC to connect with sensors and actuators.

B. Microcontroller

A microcontroller a device that can be used to control some process.[10] It includes a processor core, memory, and programmable inputs/outputs integrated in a small card. By reducing the size and cost compared to a dedicated design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control devices and processes. The microcontroller can be programmed to perform several functions and when equipped with an Ethernet card provides standard web interface.

IV. Design

After research we decided for evaporative humidifier, because it is cheap to build and to maintain. This method provides good water distribution. We decided for very efficient fan, so it can cover large area with just single humidifier. Moreover, it is environmental friendly due to its small energy consumption.

For controlling system we decided for microcontroller, due to its the price, size and low energy consumption.

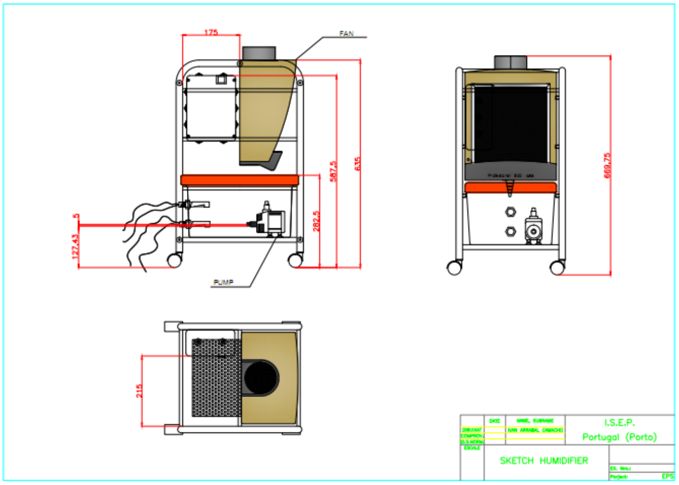

Fig. 1. Model of humidifier

Our humidifier consists of fan, pump, water tank, and cloth which is used as filter. Cloth should be wet all the time, so we decided for pump with sprays the cloth with the water from tank. The fan sucks water from cloth and then it blows as a small particles of water.

A. Fan

As a fan we used kitchen extractor. It has a metal filter and we could change it for cloth. Moreover, it is possible use the tray, which is under the kitchen extractor to flow the excess water to the tank. However, it still needed some medications. We wanted water to flow on the filter, so we made a hole in left down side to put inside the extractor a pipe with diameter 8 mm. This pipe is fixed in a top of the filter. Then we made 8 holes in the pipe to flow the water. Finally we made another hole in the middle and in front of the tray to flow an excess of water to the tank.

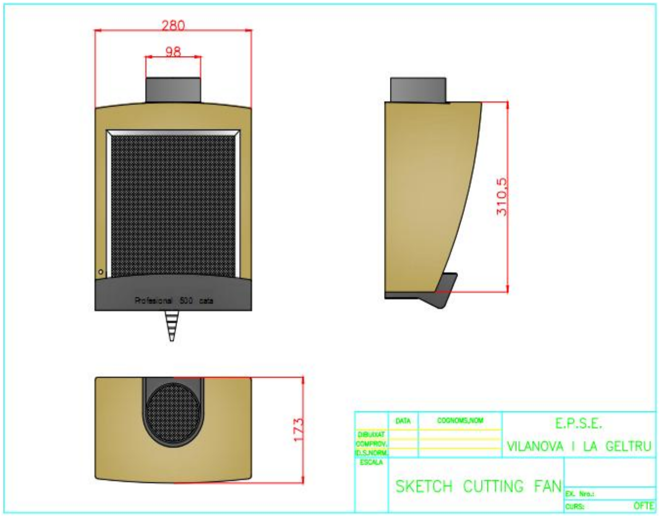

Fig. 2. Kitchen extractor

B. Water tank

In order to find the proper size of water tank, first we made a chart with data about humidity and temperature in the data center. We tried to calculate how many liters of water will be needed for the humidifier to work two days without refill. It was difficult to calculate exactly, because the temperature is not constant. However, we estimated that ten liters will be enough, even for the worst case, when the humidity is very low. Moreover, we were looking for tank with such a cover, which fit perfectly in order to prevent vaporization of water.

When we finally found the proper tank, we still had to make some changes. We made two holes for the water sensors level. Before that, we calculated the volume and we decided to make the first hole when in the tank there are 6 liters of water - when water level is below the first water sensor alarm is send. We made second hole when in the tank there are 3 liters – this is minimal amount of water needed for pump to work properly.

We made three holes in the lid of water tank - one for the water pipe from pump to the extractor (Ø 8 mm), second for pump's electrical cable, and last one, the biggest, to flow the water excess to the tank - this hole is protected with a rubber gasket.

To make refilling the tank easier, we cut the lid in two parts and joined them with a hinge and then we taped up to make it more hermetic.

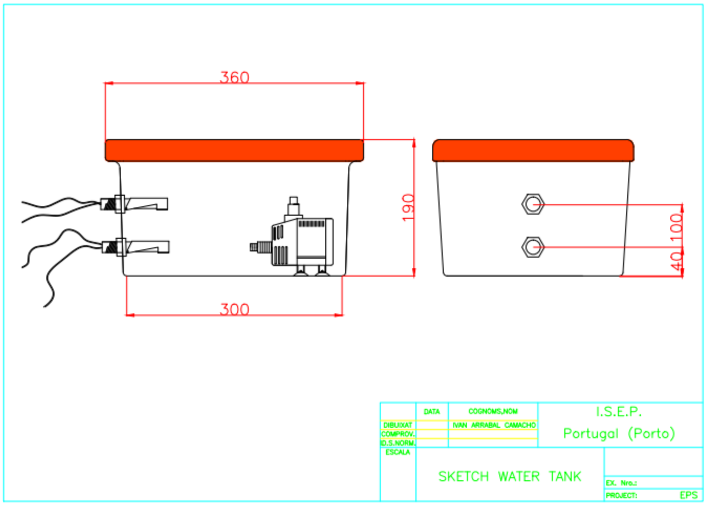

Fig. 3. Water tank

C. Structure

To set up all parts of the humidifier, we thought about a structure to organize all equipment. At the beginning we wanted to build it our own using aluminum, but then we found a bathroom furniture made of aluminum and that seemed to be perfect solution of this problem. This structure is very good for our needs. However, we had to modify it a little. We changed the metal platform in the lowest level and fixed another metal bar in the middle in order to fix the extractor. Finally, we fixed two 90º aluminum angles to fix the electrical box.

D. Filter

We were looking for materials, which could be used as a filters. We made test for all of them. For this purpose we used a hairdryer, pc fans and different materials and we measured the humidity. By this test we also proved that our humidifiers will work. During tests we considered hygroscope, how fast it dries and the differences in the humidity. Finally, we decided for professional filter for humidifiers – although it is a bit more expensive, it is much more efficient and durable, so there will be no need to change it so often.

To improve the efficiency of our humidifier we decided to mixed the professional filter and another filter from the kitchen "Vileda" (viscous fiber, polypropylene and cotton).

E. Controlling System

This humidifier is controlled by Arduino Uno. The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything what is needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

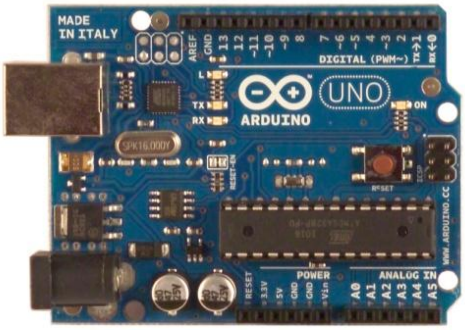

Fig. 4. Arduino Uno

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

Ethernet card is able to communicate with the Server computer and then retrieve the values of sensors and it has the ability to send data.

In accordance with customer expectations the humidifier will use the 3 currently installed humidity sensors values and values of 5 more humidity sensors which will be installed in the future and it will use one built-in sensor for that case when no connection.

Ethernet connection will be also used for sending warning emails when the water level is under the second safety level.

On the inputs there are two water level sensors and a built-in humidity sensor. A fan and a pump are connected to Arduino outputs trough relays. The warning LEDs are on the outputs too or on a parallel way of the controlled part.

When Arduino gets the data from the server, it checks the date. If a data is not available or a date is not correct, we do not use those values, we will use the lowest value of the correct values. Based on the lowest value Arduino switch on the pump for 6 seconds, then turn on the fan for 3 minutes if the humidity level is under 55%. If the humidity level is under 45% the working times change to longer. Above 55% it does not switch on again the pump and the fan, and in all cycle send a signal to relays to switch off a pump and a fan.

The water level is monitored too, by two switches. When the lowest water level sensor is closed, it means that there is not enough water in a water tank. Running without water is harmful to for a pump, therefore if there is not enough water, switches of pump and fan are blocked, and a signal is send to relays to turn off. Furthermore Arduino send an email to the chief of the data center about the water level when the level is under the second safety level.

F. Electric and electronic part

In the Arduino card there are two inputs and two outputs. We have to control the extractor and the water pump. To control it, we chose two relays. Due to the fact, that Adriano’s outputs works with 5 V, the coil needs to work also at such voltage. Because we work with maximum current that Adriano can hold, we decided to make additional electronic circuit. To activate the relay coil we used a transistor (BD137) in case, when the Arduino does not give enough power. We also used a protection diode (1N4001) to remove the current induced by turning off the coil.

We needed to calculate the resistance of the circuit

Using the transistor BD137 with hfe = 170

$$Rcoil = 55\Omega$$

$$V = 5V$$

$$I = \frac{5V}{55\Omega} = 0{,}09\ A$$

$$hfe = \frac{Ic}{Ib} => Ib = \frac{Ic}{hfe} = \frac{0{,}09A}{170} = 5{,}29 \cdot 10^{-4}A$$

$$R = \frac{Varduino}{Ib} = \frac{5V}{5{,}29 \cdot 10^{-4}A} = 9481{,}80\Omega$$

We chose one of 10 KΩ.

We have another problem because the switch is not a

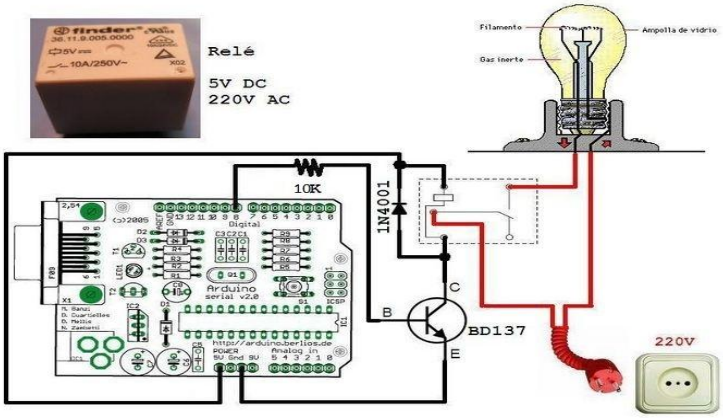

Fig. 5. Relay additional electronic circuit

commutator and the input of Arduino needs a real 0 to understand the 5V from sensor level.

We put a resistance from 10 KΩ to do the commutation.

We needed a power supply to convert 230 V AC to 12 V DC because Arduino works with lower.

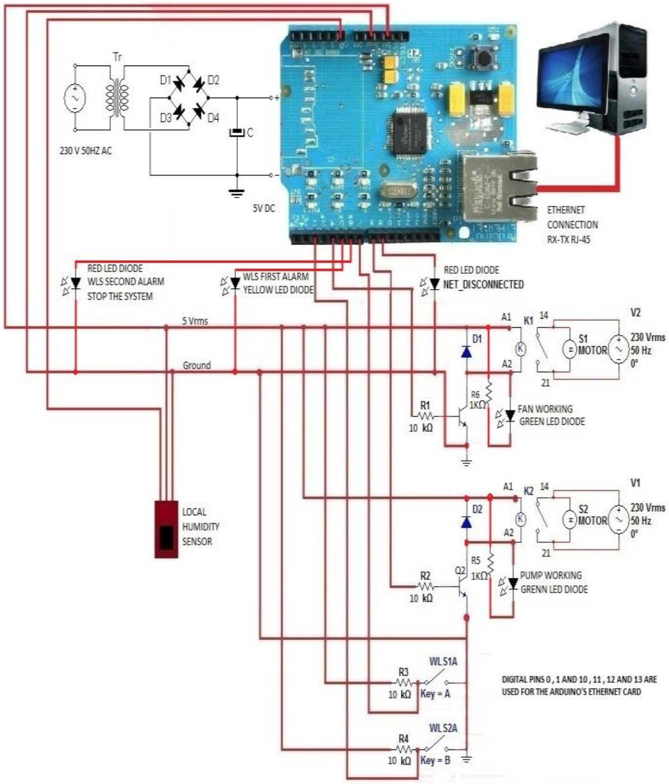

Fig. 6. Electrical and controlling sketch

G. *Programming*

Variables:

Global variables:

- const String s1 and const String s2: HTTP codes, between this two string changes the number of the humidity sensor to get humidity values from data server.
- const String s3: HTTP code to get date from the TomCat server.
- const int SENSOR_NUMBER: constant number of the humidity sensors in the data server.
- const int NOT_FOUND: constant variable t compare the reply if no data available.. Five sensors from the eight are not installed yet.
- byte mac[] : MAC address of Arduino.
- byte ip[] : IP address of arduino.
- byte server[] : IP address of the Data server
- int min_value: The minimum humidity value of the eight sensor of the data centre and the one built-in sensor.
- int water: is a flag for water level. If it's "1" the humidifier runs are allowed, if it's "0" it means not enough water for the pump and the run is banned.
- int server_minutes:
- int sensor minutes:
- boolean email: is another flag for email sending. "0": email sending is allowed. "1": email send was successful and it blocking the email sending while the water has filled up.
- String server_date: the answer when the program calls the date from the server.
- String sensor_date: part of the reply what included just the date of the sensor.

Local variable:

1. Local_Humidity():
    - int LHS: current humidity level from the built-in sensor(read from analog pin 0).
2. Water_Control():
    - int WLS1: the firs water level switcher state(read from digital pin 7).
    - int WLS2: state of the second water level switcher(read from digital pin 8).
    - String alarm: It's a message to the serial monitor about the water level.
3. SendMail():
    - Client Emailclient: availability of the email server.
4. getDataFromServer():
    - Client client: availability of the data server.
    - String extract_server_date: current date of the data server.
    - String extract_sensor: current date of eight humidity sensor.
    - float getValue: humidity level from the eight sensor.
        - int finish: end of the humidity value in the reply.
        - float res: value of the humidity (8 sensors) from the reply in float format.
        - String roi: value of the humidity (8 sensors) from the reply in string.
    - String reply: the answer of the data server that include sensor Id-s, dates, and the humidity values. From this reply we have to take out the important data.
    - int getHours: this string get the hours from the time .
        - int h: number of hours.

- int getMinutes: this string get minutes from the time.
    - int m: in the current time the number of the minutes.
- String getTimeInMinutes: thin string convert hours and minutes from the current time to minutes.
    - (1)int counter: counter 1-8.
    - (1) int nchars: number of the characters.
- getDate():
    - (1)char c: one character from the date reply, changes letter by letter.
    - (2)int nchars: number of the characters.
- getHumidity():
    - (2) int counter: counter 1-8.
    - (2) char c: character from the reply, changes letter by letter.
    - int value: eight humidity values.

Main parts of the program:

(1) Void setup:

In this part, the program set the pins to outputs or inputs. Furthermore initialize the connection to data server.

(2) Humidity control based on local humidity sensor:

Those seven lines read value from the local humidity sensor and put into the "min_value" if it is lower than the "min_value". The "min_value" variable is that variable which the program use as current humidity value.

(3) Fan and pump control:

This is a real part of humidity control, in this part we control when time turn on and off, and how long time will run the pump and the fan if the humidity level is under 55%.

(4) Connection to Data server:

"void Connection () " makes connection between the TomCat server and Arduino. If the connection was not successful, and it repeat the process again and again while not connect. For the period, while the connection is not prosperous the program will use just the local humidity sensor values.

(5) Connection to email server:

It is the same process like "void Connection()", but here Arduino connecting to the email server.

(6) Sending email:

This module is sending a water level warning email.

(7) Operation cycle

There are three more functions in the program. One is call the current date from the data server, than call the date of values and compare it. The program will ignore values with incorrect date. And in another function the program get the humidity data from the server. In the case when "no data available" or the value is 0 will ignore too.

V. Discussion

We had few problems during performing our project. When we were looking for good solution, we could not decide which type of humidifiers will best suit all needs. We were considering ultrasonic humidifier, but we realized it is very complicated and creates additional problems. So finally we decided for evaporative humidifier as it combines good humidifying without a risk of too much humidity in air and simplicity during building.

We also had to repeat testing filters, as the measurements were incorrect.

We had also some problems how to connect the electrical switchers to Arduino and we needed to create a circuit from a transistor, a 10 KΩ resistor and a protection diode, because we could not connect Arduino directly to the coil.

In further development it is possible to make coverage around the structure and change the aluminum structure to composite material structure to hold the control unit and the water container. Moreover, it is possible to add new functions based on individual needs (in program or in architecture);

.

Acknowledgment

First of all, we would like thank to ISEP for an opportunity to perform this project.

We also thank to José Barros Oliveira for answering lots of questions with endless patience.

We would like to thank to all supervisors, who helped us to deal with problems, which occurred: Maria Benedita Malheiro – for coordinating weekly meetings; João Francisco Silva and Betina Campos Neves for asking questions that led us to solve problems; Nídia Sá Caetano, Pedro Barbosa Guedes, Manuel Santos Silva – for advices and support.

Finally, we thank to all staff for friendly acceptance.

References

[1] Iallergy. Humidifier Buying Guide. Available at: http://www.iallergy.com/category11_86/default.html Accessed in March 2011.

[2] okazje.info. Kupujemy nawilżacze powietrza. Available at: http://www.okazje.info.pl/porada/dom-i-ogrod/kupujemy-nawilzacze-powietrza.html Accessed in March 2011.

[3] Buzzle.com. Types of humidifiers. Available at: http://www.buzzle.com/articles/types-of-humidifiers.html Accessed in March 2011.

[4] JS Nawilzacze. HumiPac - ceiling mounted humidifier. Available at: http://www.jsnawilzacze.pl/humipac-ceiling-mounted-humidifier-510-details/ Accessed in March 2011.

[5] JS Nawilzacze. HumEvap MC3 - nawilżacz wyparny /chłodnica. Available at: http://www.jsnawilzacze.pl/humevap-mc3-nawilacz-wyparny-chodnica-486-details/ Accessed in March 2011.

[6] Nuclearts.pl Nawilżacze powietrza. Jaki wybrać? Available at: http://nuclearts.pl/dom/nawilzacz-powietrza-jaki-wybrac-czym-sie-kierowac-przy-zakupie/ Accessed in March 2011.

[7] Projektoskop. Nawilżacze powietrza. Available at: http://www.projektoskop.pl/a-6936-nawilzacz-powietrza.html Accessed in March 2011.

[8] Media sklep. Available at: http://mediasklep.ogicom.pl/nawilzacz-powietrza-triada.html Accessed in March 2011.

[9] Wikipedia The free encyclopedia - Programmable logic controller. Available at: http://en.wikipedia.org/wiki/Programmable_logic_controller Accessed in March 2011.

[10] Wikipedia. The free encyclopedia – Microcontroller. Available at: http://en.wikipedia.org/wiki/Microcontroller Accessed in March 2011.

[11] Wikipedia. The free encyclopedia – Humidifier. Available at: http://en.wikipedia.org/wiki/Humidifier Accessed in March 2011.

[12] Engineeringtoolbox Available at: http://www.engineeringtoolbox.com/ Accessed in March 2011.

[13] AMCO Available at: www.amco.com Accessed in March 2011.

[14] John Gibson Displays Available at: www.ferret.com.au Accessed in March 2011.

[15] Bryant Available at: www.bryant.com Accessed in March 2011.

[16] Siemens Available at: www.automation.siemens.com Accessed in March 2011.

[17] Schneider Available at: www.schneider-electric.com Accessed in March 2011.

[18] Arduino Available at: www.arduino.cc Accessed in March 2011.

[19] Beyond Logic Available at: www.beyondlogic.org Accessed in March 2011.

[20] Maxim Available at: www.maxim-ic.com Accessed in April 2011.

[21] Sparkfun Available at: www.sparkfun.com Accessed in April 2011.

[22] PTrobotics Available at: www.ptrobotics.com Accessed in April 2011.

[23] MANN-FILTER Available at: www.mann-hummel.com Accessed in April 2011.

[24] Beach filters Available at: http://www.beachfilters.com/ Accessed in April 2011.

[25] Humidifier Filters Available at: http://www.humidifierfilters.com/ Accessed in April 2011.

[26] KAZ Available at: www.kaz.com Accessed in April 2011.

[27] Filters for home Available at: http://www.filters-for-home.com/Humidifier_Filters.html Accessed in April 2011.