

BLUETOOTH APPLICATION FOR TEMPERATURE AND HUMIDITY CONTROL

Emil Gocan¹

Gabriela Bucur¹

¹ Petroleum-Gas University of Ploiesti, Romania
e-mail: gbucur@upg-ploiesti.ro

DOI: 10.51865/JPGT.2023.01.22

ABSTRACT

The paper presents an application that involves the real-time acquisition of two quantities: humidity and temperature; displaying them on a high-resolution OLED screen; activation of a cooling fan when the set temperature value is exceeded; the possibility of setting the reference temperature value through the Android/iOS mobile application, which communicates with the installation via Bluetooth. To create the application, programming algorithms are used that read these quantities in real time using the DTH 11 sensor, an Arduino Uno development board equipped with the Atmega328 microcontroller, using digital and analog pins. Information is displayed on a high-resolution colour OLED screen attached to the Arduino board. From the created mobile Android application, the temperature value to be compared can be set and an LED for ON/OFF status can be ordered.

Keywords: Bluetooth communications, temperature control system, Arduino, Android/iOS mobile application

INTRODUCTION

Temperature is a physical quantity that quantitatively expresses the hotter or colder property of the respective system. Thus, the material with a higher temperature is warmer, and the one with a lower temperature, colder. Temperature is a parameter that characterizes the speed with which it moves the atoms_what make up a substance. In the case of heating a body, the speed can increase up to a certain limit. Scientists say that at an extremely low temperature, called absolute zero, atoms or molecules would stop moving completely, but reaching this value is not possible by thermodynamic means [1].

Humidity is the amount of vapour of the water contained in a sample of air. There is three ways of expressing humidity: absolute humidity, relative humidity and specific humidity. Absolute humidity is the amount of water vapour (in grams) contained in a unit volume of air. Relative humidity (abbreviated RH) is the ratio of the amount of moisture contained in the air to the maximum amount that can be contained in the same amount of air in theory. Relative humidity depends on temperature and pressure [2].

Monitoring environmental variables such as temperature and humidity has a long history. Measuring and keeping these environmental variables constant is important in industrial processes.

TEMPERATURE AND HUMIDITY CONTROL SYSTEM DESIGN

The diagram of the system for measuring and displaying humidity and temperature values, with the possibility of controlling them via Bluetooth, is presented in Figure 1.

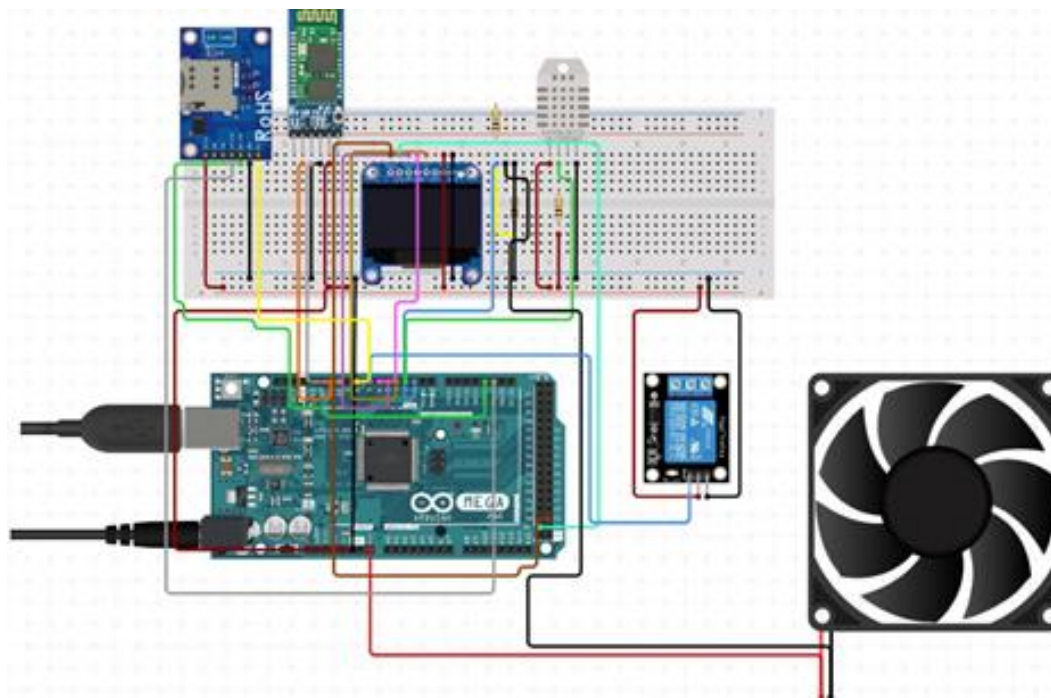


Figure 1. Temperature and humidity control system connection diagram [2]

The structure of the system includes the temperature and humidity sensor DHT11, the Arduino Uno development module, an OLED screen, a Mit App Inventor module, the Bluetooth communication module, the cooling fan when the set temperature value is exceeded. These components are presented in detail below:

DHT11 is a high-performance humidity and temperature sensor (Figure 2). The sensor provides good reliability and stability. Temperature is measured by an NTC thermistor and relative humidity is measured using a capacitive sensor. The output is provided as a digital signal [2].

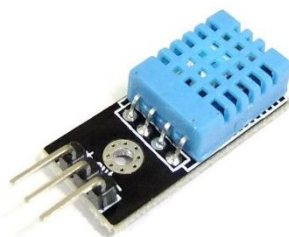


Figure 2. Humidity and temperature sensor [2]

OLED screen: This is a high-contrast OLED display (Figure 3) with a resolution of 128x64 pixels. Display control is done via the SPI interface [2].

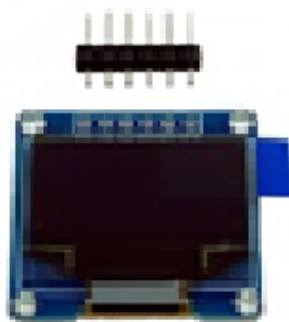


Figure 3. OLED Screen [9]

Bluetooth module HC-05: can be used in projects where you need to transmit data very easily, but not via cable, at reasonable distances of up to 10m (Figure 4) [2].



Figure 4. Bluetooth module HC-05 [2]

5V controlled relay: it is useful for various IoT (Internet of Things) projects, with the help of which we can automate the projects. Devices that operate at 230V AC or 30V DC and consume up to 10A can be controlled (Figure 5)[2].



Figure 5. Bluetooth module HC-05 [2]

ARDUINO development board. Arduino is a programmable microcontroller board [7] through which it is possible to create and develop objects that interact with the necessary programming software (Figure 6). The Arduino IDE (Integrated Development Environment) software provides space to write code in the language that the Arduino board understands and responds to. The programming languages used are C and C++. Low price, easy-to-use design, and flexibility for hardware and software modifications are some of the characteristics of microcontroller-based Arduino hardware and software [6].



Figure 6. ARDUINO development board [6]

ARDUINO CODE IMPLEMENTATION

After making the connections between the elements, as shown in Figure 1, we move on to writing the code for the Arduino board, in a manner similar to the C language. From the captures attached below we can see the structure and code of the Arduino.

```
temp_controlled_fan_v2
#include <U8glib.h>
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);
#include <DHT.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include <string.h>
|
#define DHTPIN 4 // pin senzoz DTH
#define ONE_WIRE_BUS 5
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature tempsensor(&oneWire);
```

The program requires access to several libraries, such as the U8glib.h library which is used to control the OLED screen; another necessary library is DHT.h through which we have access to the data read by the temperature sensor. The OneWire.h and DallasTemperature.h libraries were also used in the tests, which also use another type of temperature sensors with a higher precision; the SoftwareSerial.h library contains directives to access and create objects that use the serial communication interface and through which we can designate the pins in the Arduino interface that are connected to sensors and modules (for example the Bluetooth module). A great advantage of the OLED

screen is that it allows the display including graphic elements, we can specify the font used, the message, etc.

After defining the pins connected to the sensors and the communication modules, the communication speed is initialized, the routines are created to set and listen to the information received through the Bluetooth module.

The main program void loop(){} is the loop where the code is written that runs without stopping, in which it is checked if there is a Bluetooth connection and communication, the received data is analysed and commands are given to various peripherals or devices to start working: for example , we turn on the LED, we command the relay connected to the fan to turn on, if the temperature sensor detects that we have exceeded the set temperature, etc.

```
float hum = dht.readHumidity();
float temp = dht.readTemperature();
//tempsensor.requestTemperatures();
//float temp2 = tempsensor.getTempCByIndex(0);

void loop() {
  if(Bluetooth.available() > 0)
  {
    recvWithEndMarker();
    if (newData == true) {
      dataNumber = 0;
      dataNumber = atoi(receivedChars);
      Serial.print("receptionat ... ");
      Serial.println(receivedChars);
      Serial.print("Data ca numar ... ");
      Serial.println(dataNumber);
      if(dataNumber == 1)
        digitalWrite(13, HIGH); //aprint led
      else if(dataNumber == 0)
        digitalWrite(13, LOW); //sting led
      else
        tempMin = dataNumber;

      newData = false;
      //memset(receivedChars, 0, sizeof receivedChars);
    }

  } else {

    char tempMinVal[10];
    char humVal[10]; //pt conversie float in char
    char tempVal[10]; //pt conversie float in char
    char tempVal2[10];

    // picture loop
    u8g.firstPage();
    do {
```

All this received and set information is displayed on the high-resolution OLED screen so that the user is informed about the current state of the system.

ANDROID/IOS MOBILE APPLICATION

In order to create the mobile application to run on smart phones [5], although other programming languages can also be used (eg Xamarin from Visual Studio), the Mit App Inventor platform was used, efficient enough for the intended purpose [8].

It is a platform where we can create much faster code for mobile devices, using already predefined libraries and command blocks (Figure 7).

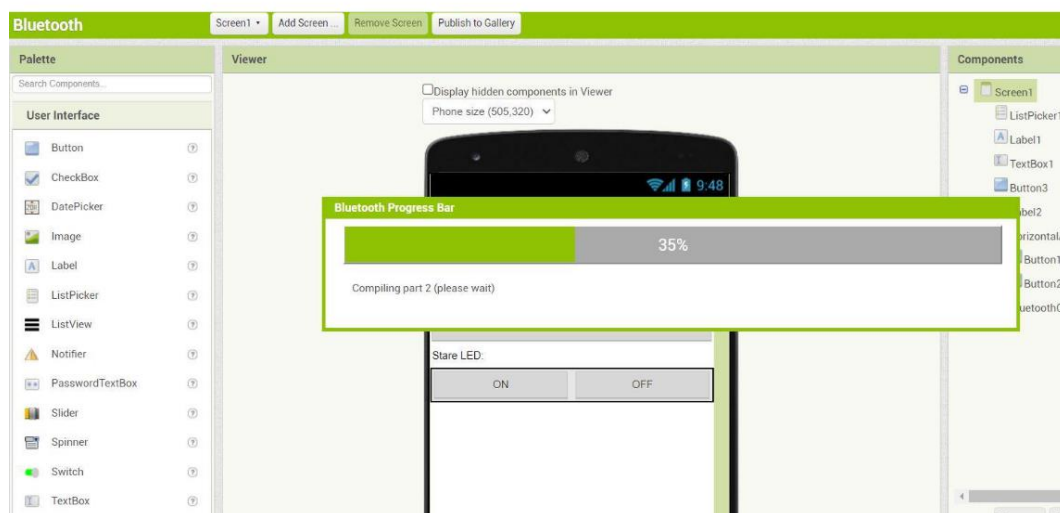


Figure 7. Android mobile application for temperature and humidity control system [8]

EXPERIMENTAL RESULTS

In testing the application, we used the mobile interface created through which we connect to the Bluetooth module, we receive the message "Connected" which confirms that we are connected.

To set the reference temperature around which we order the cooling fan to turn on or off, we have a textbox in which we write the temperature and a button through which we call the "Send Text" function. of the Bluetooth module.

We have also created ON/OFF buttons through which we can turn on or turn off a led that is connected to the Arduino interface.

When put into operation, the Arduino interface starts with a preset temperature threshold, but after setting the reference temperature in the mobile application, this threshold changes and is displayed on the OLED screen, together with information about the values read by the sensors and the command status of the ventilator (Figure 8).

During the tests, a GSM communication module was also used to send information via SMS about the current state of the system, but there were some problems in establishing communication via the SIM card.

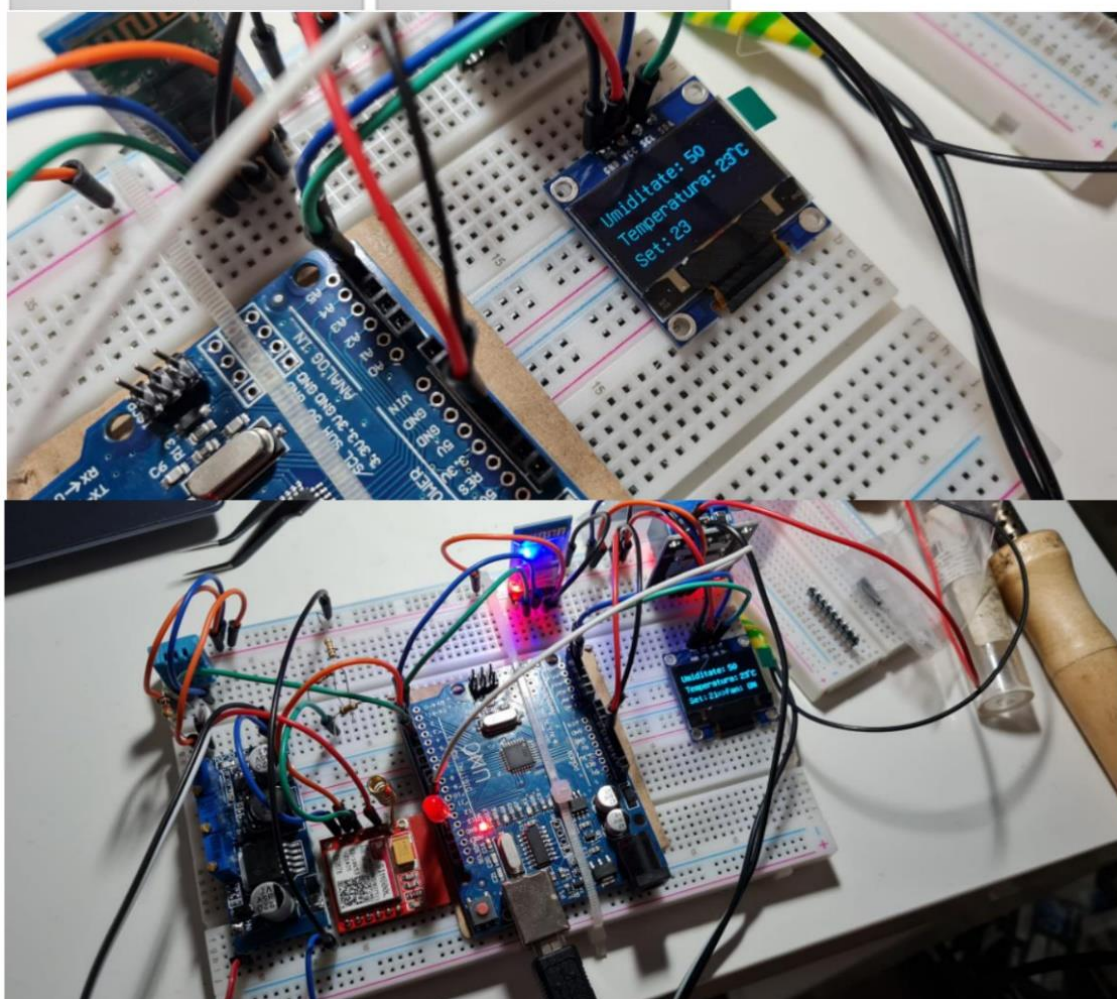
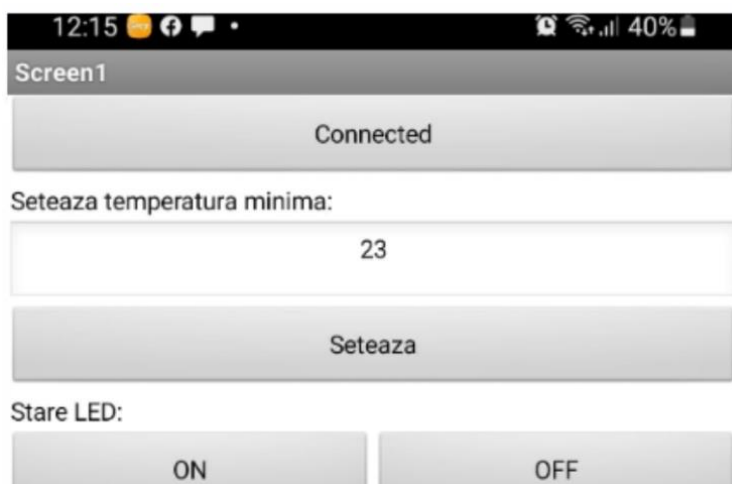


Figure 8. Temperature and humidity control system implementation

CONCLUSION

The presented application is the basis of any automation through which we want to monitor the temperature in a room and, depending on the set temperature, to order heating or cooling equipment. Since the control relay can also be used to control equipment with



230V power supply, the assembly presented can also be successfully used in projects outside the didactic sphere. In addition, it was desired to increase the degree of complexity by using Bluetooth or GSM communication technologies, all controlled by a mobile application, capable of running on the latest generation smartphone devices.

By using the GSM interface, we can always have real-time data about the state of the system and through a simple integration with a data storage web service on a cloud server, we can also have statistical data for later consultations.

Also, through the GSM module, we could set the system from a distance, not just from the vicinity of the equipment, which at the moment is limited to 10 m due to the limitation of Bluetooth communication.

REFERENCES

- [1] Bucur, G., *Practica automatizarii proceselor industriale*, Editura UPG Ploiești, 2017, pp 85-99
- [2] Gocan, E., *Proiectarea si realizarea unui sistem de masurare a temperaturii si umiditatii, proiect la disciplina Masurari si Traductoare*, coordonator conf. G. Bucur, UPG Ploiesti, 2022
- [3] Jaganmohan Reddy K., Santhosh Kumar B., *Patient Temperature Monitoring System Using Bluetooth Communication*, *International Journal of Engineering Trends and Technology (IJETT)*, Volume 16, Number 4, pp. 147-149, Oct 2014.
- [4] Magalhaes, A.C.M., Mestria, M., *Temperature Control System using Mobile Application Interface*, *European Journal of Formal Sciences and Engineering*, January-June 2022, Volume 5, Issue 1, DOI: 10.26417/729pbt84
- [5] Mohammed, J., Srija, U., *Bluetooth Based Remote Monitoring & Control System*, *Journal of Basic and Applied Engineering Research*, Volume 1, Number 8; October 2014 pp. 108-111
- [6] <https://docs.arduino.cc/hardware/uno-rev3>
- [7] <https://forum.arduino.cc/t/gsm-shield-sms-receiving-without-serial-monitor/481721> <https://www.arduino.cc/reference/en/libraries/u8glib/>
- [8] <https://appinventor.mit.edu/>

Received: April 2023; Revised: May 2023; Accepted: June 2023; Published: June 2023