

The climate control system using ESP8266 and Arduino IoT Cloud

Serhii Tsyryllyk^{a,b}, Maksym Tsyryllyk^c, Nadiia Potapova^c, Andriy Semenov^d and Volodymyr Tromsyuk^a

^a Vinnytsia Technical Applied College, Khmelnytske highway, 91/2, Vinnytsia, 21021, Ukraine

^b Vinnytsia National Agrarian University, str. Sonyachna, 3, City, Vinnytsia, 21008, Ukraine

^c Vasyl' Stus Donetsk National University, str. 600-richchia, 21, Vinnytsia, 21021, Ukraine

^d Vinnytsia National Technical University, Khmelnytske highway, 95, Vinnytsia, 21021, Ukraine

Abstract

Humidity in the air and temperature have a long-term and destructive effect on people, machines, and materials. This paper proposes a system designed for remote control and monitoring of the temperature and humidity of the premises of an educational institution. To increase the accuracy of temperature and humidity measurement, the method of weighted moving average is used. Such a system allows you to determine the dew point or thermodynamic parameters of the air to control ventilation, heating, air conditioning in the room, during working and non-working hours, as well as to control emergencies.

This technique uses a wireless control module with an ESP8266 microcontroller using a Wi-Fi channel, which is integrated into the Internet of Things system. The microcontroller allows you to measure temperature and humidity, send the measured value via a Wi-Fi network to the IoT cloud, and perform remote control of the load, which is connected to the 220 V power grid through a relay module. The free cloud service Arduino IoT Cloud allows you to collect, analyze and manage data from a mobile application IoT Remote and web dashboard. The proposed system can be improved and integrated into the centralized climate control system of the educational institution, which in turn will allow the efficient use of financial resources.

Keywords 1

Climate control, remote control, data monitoring, cloud service, module IoT, temperature and humidity sensor, dewpoint, method of weighted moving average

1. Introduction

The Internet of Things (IoT) is an engineering solution in which electronic devices are combined into a single integrated system and the Internet is used as a communication channel between them. IoT platforms are used by suppliers and manufacturers of smart devices to equip their products with functions of remote control, real-time monitoring, setting up alerts and notifications, and integration with smartphones and other devices.

Temperature and humidity are critical parameters for many modern industries because of their long-term and destructive effects on materials, equipment, and people.

Special requirements for temperature and air humidity are relevant for the storage of medical and food products. Control of storage temperature and air humidity is important for various technological processes.

ITTAP'2022: 2nd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2022, Ternopil, Ukraine

EMAIL: sovm@ukr.net (S. Tsyryllyk); maximtsyryllyk@gmail.com (M. Tsyryllyk); potapova.nadin@gmail.com (N. Potapova); semenov79@ukr.net (A. Semenov); 2013tvd@gmail.com (V. Tromsyuk)

ORCID: 0000-0002-5703-9761 (S. Tsyryllyk); 0000-0003-4566-4102 (N. Potapova); 0000-0001-9580-6602 (A. Semenov); 0000-0001-5022-8159 (V. Tromsyuk)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

Air temperature and humidity are of particular importance for the transportation of fresh and frozen products; greenhouses for growing plants, incubators, and premises for keeping animals.

A violation of the conditions of production, storage, and transportation of products leads to significant financial losses, and in the case of medical drugs - poses a threat to people's health and lives.

IoT-based temperature and humidity monitoring effectively reduce the cost of human resources for temperature and humidity control.

One of the important tasks is to monitor the temperature and humidity of the environment without restricting the movement of the user.

To ensure comfortable working conditions in the cold period, a system is needed that would allow assessing the state of temperature, and humidity in premises for archival storage of documents, educational institutions, pharmacies, museums, libraries, offices, and warehouse premises. This will make it possible to effectively use financial resources and, upon the improvement of this system, to automatically set the temperature on the premises, during working or non-working hours, and monitor the state of emergency.

The main problem for the premises is the high relative humidity of the air and, as a result, the condensation of moisture vapor on cold surfaces, which causes corrosion, rotting of materials, and the formation of fungal mold.

The dew point is the temperature at which condensation occurs. The dew point temperature is determined by the temperature and relative humidity of the air. The approximate calculation of the dew point T_r (°C) depending on the air temperature T (°C) and its relative humidity Rh (%) is :

$$T_r = \frac{b \left(\frac{aT}{b+T} + \ln(Rh) \right)}{a - \left(\frac{aT}{b+T} + \ln(Rh) \right)}, \quad (1)$$

where $a = 17.27$, $b = 237.7$.

During the development and operation of automated ventilation and air conditioning systems, thermodynamic analysis of air processing is carried out based on the values of the main thermodynamic parameters - heat and moisture content of moist air. The coordination of external and internal parameters in rooms with ventilation can be ensured by changing the heat and moisture content of the air [1]. Of the known methods, heat content can be most accurately determined by temperature and relative humidity.

2. System Architecture

The climate monitoring system is a software-hardware module for wireless control using a Wi-Fi channel, which is integrated into the Internet of Things system [2, 3] and allows you to measure the temperature and humidity in the room and transmit the measured values via the Wi-Fi network with remote access from a cell phone, tablet, or personal computer and performs remote control of an electrical appliance (air conditioner or heater) in manual or automatic control mode depending on the given cooling/heating algorithm.

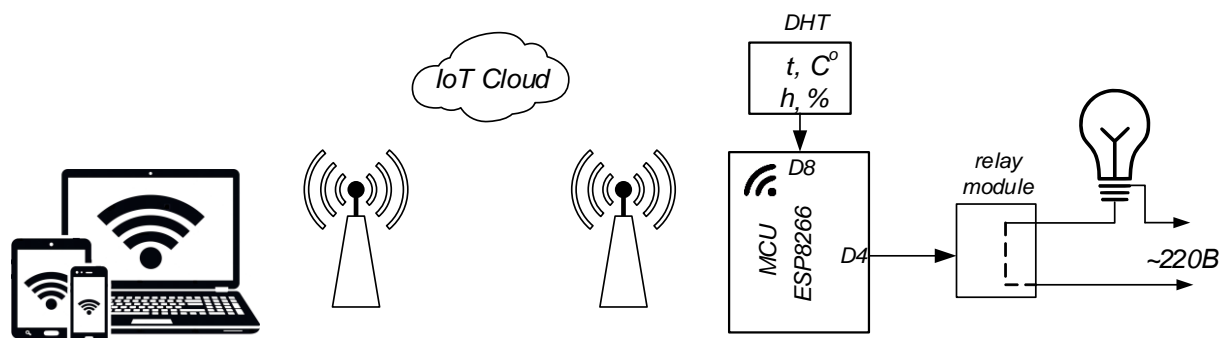


Figure 1: Structural diagram of the climate monitoring system

To exclude the influence of ambient temperature on the results of measuring relative humidity, it is recommended to use humidity sensors together with temperature sensors [1], for example, DHT11, DHT22, SHT11, SHT20, SHT30, BME280, GY-21, AM2320.

The climate monitoring system consists of a temperature and humidity sensor DHT11 [4], a control unit with Wi-Fi network support (ESP8266 microcontroller) [5-7], an executive device - relay module, power supply unit (Figure 1). Appointment indicators: supply voltage - 220 V \pm 10%; switched load power 2500 W (10 A/250 V); the determination of temperature 0–60 °C with an accuracy of 2%; determination of humidity 20%-90% with an accuracy of 2%; support for Wi-Fi networks of the 802.11 b/g/n standards.

2.1. Hardware

The main element of the IoT-based monitoring and control system, which receives information from the DHT11 sensor and transmits it via Wi-Fi to a cloud service that the user can join and view temperature and humidity information and remotely control devices that can affect climatic conditions of the control object: heating or cooling, humidification or drying of the air.

Due to the ease of programming and ease of integration into various embedded systems, Arduino boards are among the most common for electronic projects. The first generation of Arduino boards were general-purpose microcontrollers that connected to the Internet using GSM and Wi-Fi modules, but as the concept of the Internet of Things began to develop, boards with special functions supporting the Internet of Things began to be developed: Arduino 101, Arduino MKR1000, Arduino Yun, Arduino NANO 33 IoT and others.

Chips and boards based on the ESP8266 have become one of the best chipsets for Wi-Fi-based IoT devices. These modules are quite cheap, consume little electricity, and are easy to use. Following the ESP8266 module, Espressif offers the ESP32 module, which is based on a dual-core microcontroller and has a wider set of functionality compared to the ESP8266 module. ESP chips have great flexibility and can be used as Wi-Fi modules, connected to other microcontrollers, or used in stand-alone modes without the use of additional microcontrollers. They have small form factors and simplify the implementation of functions necessary for the IoT sphere, such as, for example, firmware updates "over the air" (OTA technology - over-the-air). The availability of development boards such as NodeMCU and other ESP-based third-party boards allows developers to familiarize themselves with the capabilities of this platform before using it in projects. The ESP8266 boards come to FCC and CE certified to reduce overall device certification costs after manufacturing. ESP boards provide one of the most reliable, dedicated Wi-Fi interfaces that include several protocols that support the Internet of Things, such as the ESP Touch protocol.

ESP8266 boards are fairly easy to learn and can be used with any microcontroller to create IoT projects.

The port allocation of the WeMos D1 mini-module is shown in Figure 2. This module contains 11 general-purpose ports. Some ports have additional functions: GPIO1, GPIO3 – UART, GPIO5/D1, GPIO4/D2 – I²C/TWI, D5..D8 – SPI, D1..D7 – outputs with PWM (PWM), A0 – analog output from ADC. A temperature and humidity sensor is connected to the module through D8 (GPIO15).

To connect this module, it is sufficient to supply +5V supply voltage.

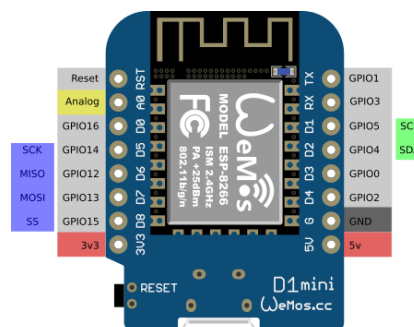


Figure 2: A pinout diagram of a module WeMos D1 mini and their corresponding functions

The key component of the monitoring system is the combined temperature and humidity sensor DHT11 (DHT22, HDC1080, or BME280). The DHT11 sensor is a digital temperature and humidity sensor. It consists of a capacitive humidity sensor and a thermistor. Also, the sensor contains an ADC for converting analog values of humidity and temperature. DHT11 outputs a calibrated digital signal. It uses exclusive digital signal acquisition and moisture sensing technology, ensuring its reliability and stability.

The sensor connection diagram of the ESP8266 microcontroller is shown in Figure 3.

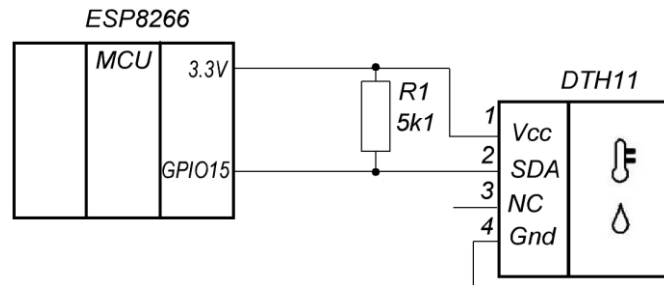


Figure 3: Connecting the DHT11 sensor to the ESP8266 microcontroller

Sensor data consist of integral and decimal parts for both relative humidity (RH) and temperature. The data from the DHT11 sensor consists of 40 bits, and the format is as follows: 16 bits of humidity data (high and low), 16 bits of temperature data (high and low), and 8 bits of checksum data (Figure 4).



Figure 4: DHT11 sensor data format

To check the accuracy of the obtained data, it is necessary to add all the values of humidity and temperature and check whether the sum is equal to the value of the control sum:

$$00100101 + 00000000 + 00011001 + 00000000 = 00111110$$

To initialize data transmission, the host device (microcontroller) must send a start signal by setting a low logic level on the data bus (SDA), with a duration of 0.8 . . . 20 ms. After sending the start signal, the microcontroller must release the data bus (resetting the GPIO line to input) and proceed to poll its status. In response to the start signal, the sensor sequentially sets low and high logic levels on the data bus, each that is 80 us in duration. Next, the sensor outputs 5 information bytes (40 bits) containing humidity and temperature data. Bytes are transmitted standardly, most significant bit first. First, the higher byte of humidity (Humidity high) is transmitted, then the lower byte of humidity (Humidity low), then the higher (Temperature high) and lower (Temperature low) bytes of temperature, and the last byte contains the checksum (Parity bit). In this way, humidity and temperature are represented by two bytes, for conversion, you must divide these numbers by 10. For example, if you received a two-byte number 638, divide it by 10, and you obtain the required value of 63.8.

The minimum interval for reading data from the sensor should be 2 seconds, with frequent access the measurement error increases. The maximum cable length is 30 m at a supply voltage of 5 V, and at a voltage of 3.3 V, the cable length should not exceed 1 m.

A Pull-up resistor must connect to the sensor (Figure 3). The sensor was always in sleep mode when the line was high. To communicate with it, we must activate it, get confirmation and accept 40 bits of data.

It is necessary to consider how to connect the executive device to the ESP8266 module. To control external devices, the same MK input/output ports that work for output are used. Signals from any line of any port can easily be used to turn on and off an external device. It is only necessary to amplify the control signal by the power to the required level. Various coordination schemes are used for this purpose. The choice of the scheme depends on the type of executive device. In the simplest case, you can use a transistor key. To control links with a large current, you must use a more powerful transistor or a complete transistor assembly. When selecting a transistor, we consider that the maximum

permissible load current for any MK output should not exceed 20 mas. If the GPIO2 port (Figure 5) is set to a single level, the switch opens and the load is connected to the power source. At a zero level at the same output, the switch is closed, and the load is disconnected.

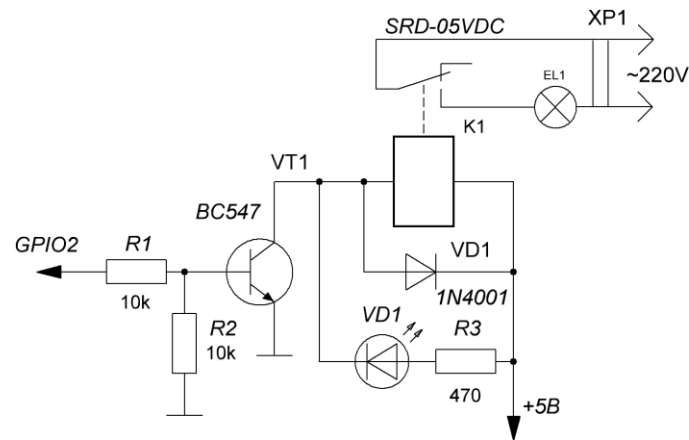


Figure 5: An executive device using a relay

If the executive device to be controlled by the ESP8266 is powered by 220 V AC, a galvanically isolated control scheme must be used. One of the possible options is a relay control scheme. A typical version of the control circuit using a relay is shown in Figure 5. The circuit shows an electronic key, the load of which includes the electromagnetic relay K1. The ESP8266 module can turn the electromagnetic relay on and off using a key. Relay contacts, in turn, control the load. Such a scheme provides switching of a sufficiently large voltage and current.

The galvanic separation between all circuits and the 220 V power network ensures the safety of working with this scheme. Diode VD1 is designed to protect the elements of the circuit from the voltage of self-induction that occurs in coil K1 at the moment of closing the switch VT1. When choosing an electromagnetic relay, one should pay attention to the following parameters. The first is the voltage of the relay. For the device under development, it should be equal to 5 V. Second, the maximum allowable switching current and the maximum allowable voltage should be for the executive contacts of the relay. They must correspond to the actual values of the current and voltage in the load circuit. We chose the relay type SRD-05VDC-SL-C.

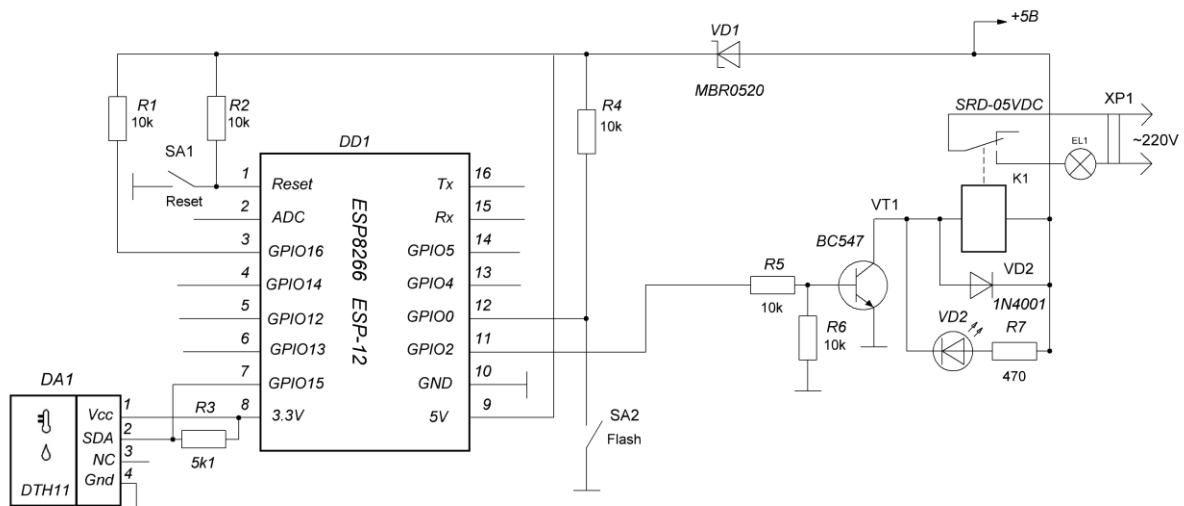


Figure 6: The circuit of the climate monitoring system using ESP8266

AC-DC voltage converter Hi-Link HLK-PM01 with an output voltage of 5 V and a maximum output current of 0.6 A is used to power the climate monitoring system with a voltage of +5 V from the mains

voltage. The block has a small-sized hermetic case with legs for mounting on a printed circuit board. There is built-in short circuit protection.

When building many measuring devices, the problem arises of reducing random errors during measurement. Increasing the accuracy of measurements is achieved by reducing random measurement errors using the weighted moving average method with a digital recursive filter, which is implemented by software [8].

2.2. Software

2.2.1. Arduino IoT Cloud

An IoT platforms are used by suppliers and manufacturers of smart devices to equip their products with functions of remote control, real-time monitoring, setting up alerts and notifications, and integration with smartphones and other devices.

IoT platforms provide seamless integration of different hardware using communication protocols, applying different types of topology (direct connection or gateway). The IoT platform is a key tool for developing IoT applications and services, which combine physical objects (things) and the network (Internet).

The most popular software IoT platforms are Microsoft Azure IoT, Amazon Web Services (AWS) IoT, Google Cloud, ThingWorx IoT, IBM Watson, Artik from Samsung Electronics, Cisco IoT Cloud Connect, Salesforce IoT Cloud, ThingSpeak, and many others [2, 3, 6, 7]. Among the protocols used by IoT platforms, the most popular are MQTT, CoAP, HTTP/HTTPS, AMQP, XMPP, and DDS [5].

Arduino IoT Cloud data storage greatly simplifies data storage and provides a full range of services within the concept of the Internet of Things (IoT). It includes hardware and software, cloud services, and a knowledge base. This cloud-based data warehouse allows users to collect, analyze and graph data from sensors, event handlers, and more. The Arduino IoT Cloud platform allows users to write the codes of their programs, compile and download them to boards directly from the browser, connect IoT devices, and use a virtual dashboard in real-time. The Arduino IoT Cloud platform automatically generates a sketch according to the settings specified by the user when the device is connected to the cloud. Arduino IoT Cloud can be integrated with Amazon Alexa, Google Sheets, IFTTT, and ZAPIER allowing users to program and control devices using voice, spreadsheets, databases, and automate notifications using webhooks. The platform also allows developers to build their applications using the Arduino IoT API, adding custom webhooks to the endpoints for greater flexibility.

When using Arduino IoT Cloud for free, one can add only two devices to the platform and the compilation time is limited to 200 seconds per day. To expand the possibility of using the platform, you can choose one of the paid tariffs, for example, ENTRY (\$2.99/month), which allows you to create 10 Things, unlimited compilation time, and unlimited storage for sketches [9].

To work with Arduino IoT Cloud [10, 12], you must create and log into an account and add a new device to it, namely, the NodeMCU ESP8266. To do this, go to the "Devices" tab and click "Add device" (Figure 7).

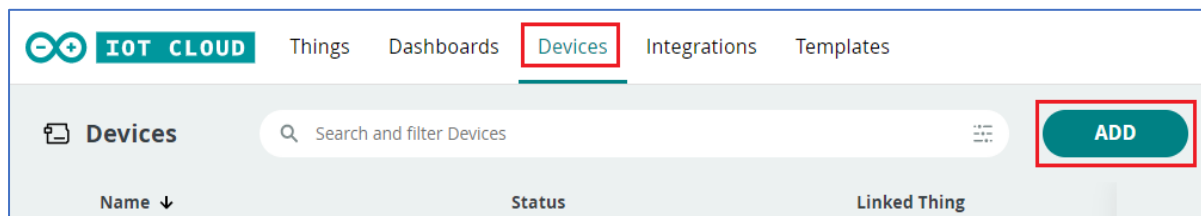


Figure 7: Adding a new device to the project

Figure 8 shows the hardware platform selection window. We press "Set up a 3rd party device" and select the development board we use. To do this, first select ESP8266 as the device type, and then select NodeMCU1.0 from the drop-down menu.

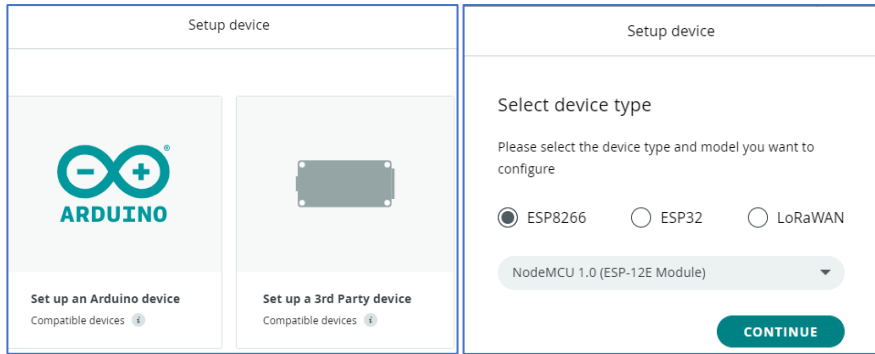


Figure 8: NodeMCU ESP8266 board selection tab

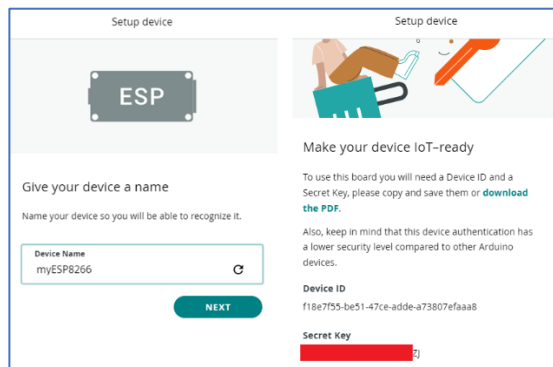


Figure 9: NodeMCU ESP8266 board selection tab

Next, set the name of the device and click Next (Figure 9). A window will appear with the device ID (Device ID) and the secret key (Secret Key) for this device. Device ID and Secret Key must be saved in notepad or click on “download the PDF” to save them in PDF format.

The next step is to create a subject/thing. To do this, go to the “Things” tab and click on “Create” (Figure 10).

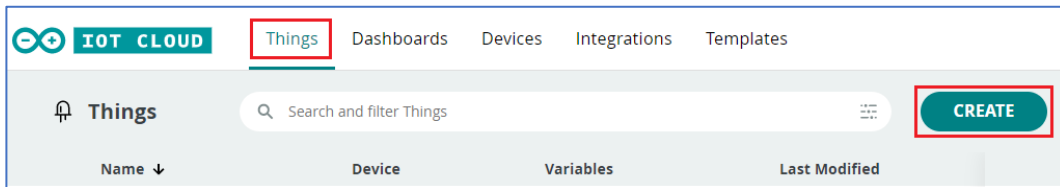


Figure 10: Things creation tab

On the Things tab (Figure 11), three options need to be configured: Variables, Device, and Network.

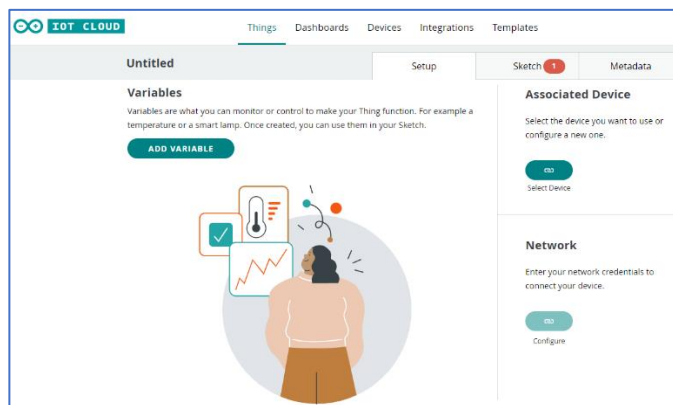


Figure 11: Adding project variables

To add variables that will be used in the program code, press the "Add variable" button. After that, a window will open in which you must fill in the information about the variables (Figure 12). In this project, we will use three variables, one of which will be used to control the relay, and the other two will be used to store the temperature and humidity values.

The screenshot shows three instances of the 'Add variable' configuration form. The first instance is for a variable named 'led'. Its data type is 'Light' (with a dropdown menu open showing options like 'Basic types', 'Light and color', etc.). The declaration is 'CloudLight led;'. The permission is 'Read Only' and the update policy is 'On change'. The second instance is for 'temperature' with data type 'Temperature Sensor (°C)' and declaration 'CloudTemperatureSensor temperature;'. The third instance is for 'humidity' with data type 'Relative Humidity' and declaration 'CloudRelativeHumidity humidity;'. All three have 'Read Only' permissions and 'On change' update policies.

Figure 12: Setting project variables

First, let's create a relay switch (led). Its data type will be Boolean (bool), its value can only be read (Read-only), and the update option (update policy) will be enabled for it. After that, click the "Add variable" button. Similarly, we add a variable for storing temperature and humidity values. The data type for these variables float, they can be read (Read-only), and the updated option for them will also be enabled. Let's associate devices with things. To do this, click the link icon in the "Device" section. After that, a window will open in which the previously created device will be available for selection. In the next step, you must enter data to access the Wi-Fi network (its name and password) and the secret key that was issued during the installation of the device. To do this, click the Configure button in the Network section and enter the necessary information (Figure 13).

The screenshot shows the 'Variables' section with a table containing three variables: 'humidity' (CloudRelativeHumidity), 'led' (CloudLight), and 'temperature' (CloudTemperatureSensor). The 'Associated Device' section shows a device named 'myESP8266' with ID 'f18e7f55-be51-47ce-adde-4...' and status 'Offline'. The 'Network' section has a 'Configure' button highlighted with a red box.

Figure 13: Setting up data to connect to Arduino IoT Cloud

The next step is to create a dashboard. To do this, go to the Dashboard tab and click on Create (Figure 14).

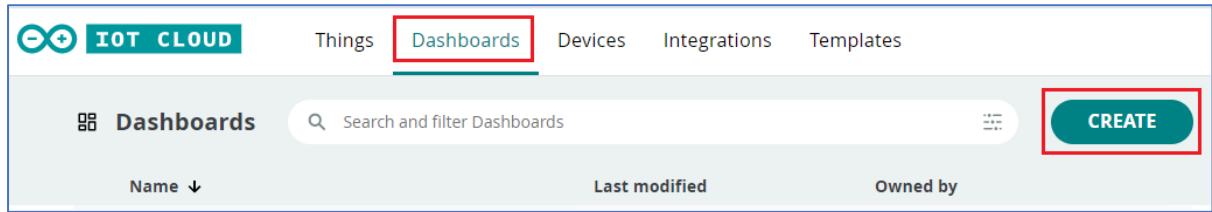


Figure 14: Creating a project dashboard

To add widgets to the dashboard, click on the pencil icon in the upper left corner, then click the “Add” button. We go to the “Widgets” tab and add the “Switch”, “Percentage Gauge”, and “Chart” widgets, which we associate with the led, humidity, and temperature variables (Figure 15). In the upper right corner of each widget, there is a context menu with the “Edit setting” command, which allows binding to variables, a range of values, and an initial value. Therefore, Figure 16 shows the settings of the properties of the “Chart” temperature change diagram.

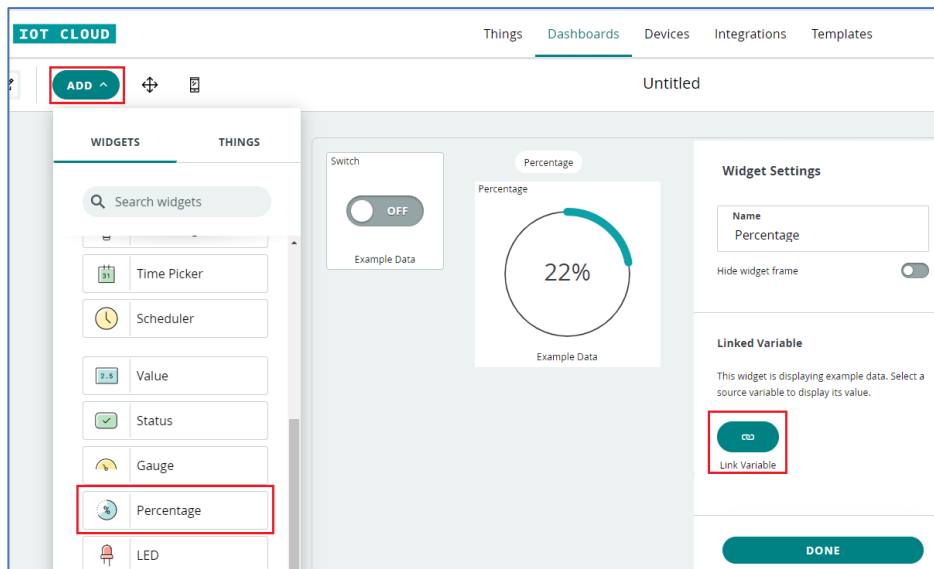


Figure 15: Project dashboard settings

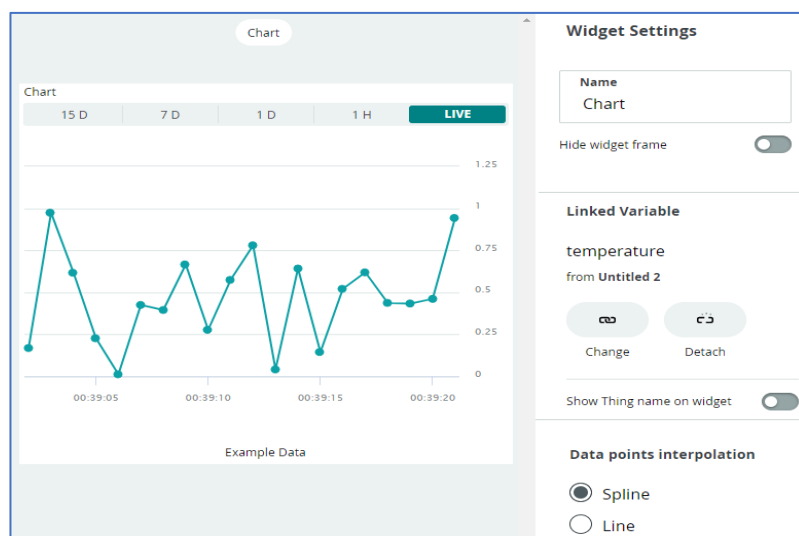


Figure 16: Chart widget settings

The software consists of a program for the ESP8266 microcontroller and provides for its interaction with mobile service or online service. When choosing, you must decide what functionality the user wants to get: the Wi-Fi module is an access point and data can be obtained only by establishing a connection with the module via a wireless channel; the module is connected to a Wi-Fi network and transmits data to a mobile application and/or to the cloud, which will allow separate control and monitoring of data; temperature and humidity information is private or shareable; data monitoring only from mobile devices and/or personal computers; how often to receive data; need to keep the data and for how long.

2.2.2. IoT module software

After configuring the Arduino IoT Cloud platform, it is necessary to program the NodeMCU ESP8266 board to read data from the DHT11 sensor, transmit it to the cloud, and perform remote control functions.

When adding variables to “Things”, the sketch in the cloud is automatically updated. Most of the application code (sketch) is automatically generated by the platform. It is necessary to add lines to work with the DHT11 sensor and implement the remote control algorithm. To do this, open the Arduino Cloud Editor code editor on the Sketch tab (Figure 17).

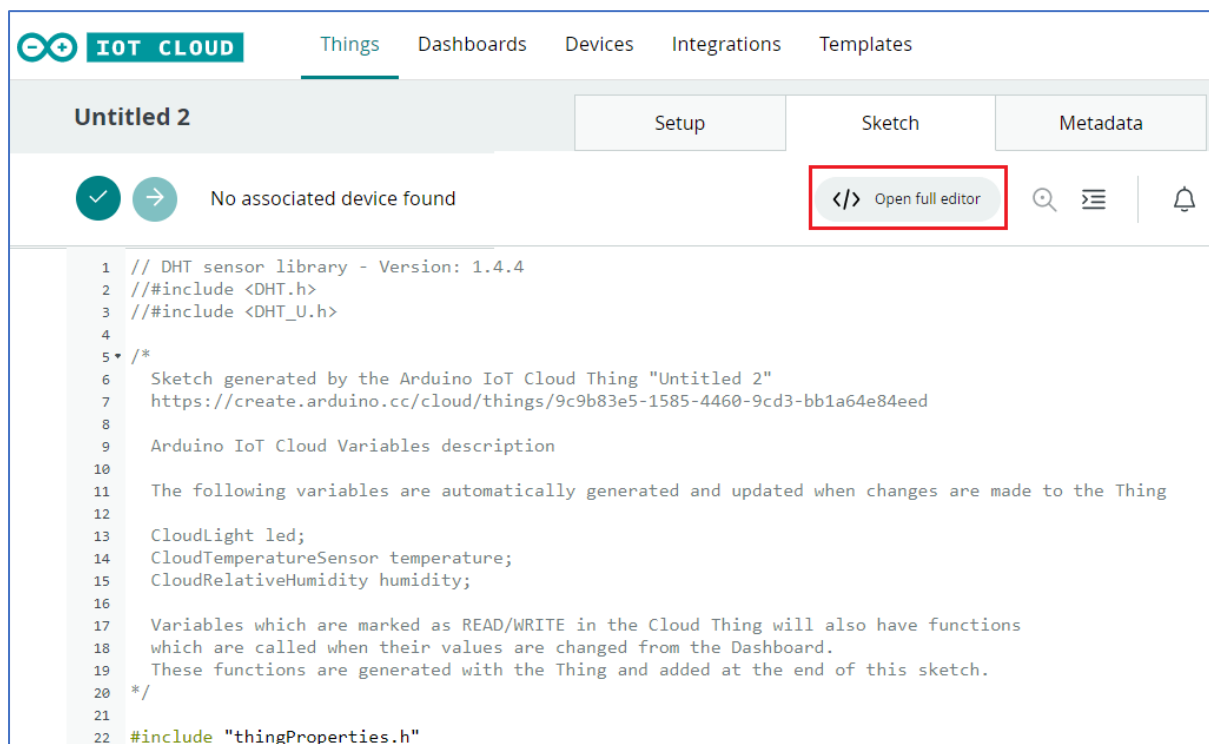


Figure 17: Editing the sketch code automatically generated by the Arduino IoT Cloud platform

The DHT.h library is used to work with the DHT11 sensor [4]. The DHT.h library has built-in functions *readTemperature()*, *readHumidity()* for reading temperature and humidity values. The command is used to connect the library

```

#include "DHT.h"
#include <math.h>

```

We create an object of the DHT class *dht* (*DHTPIN*, *DHTTYPE*). In the first parameter, we specify the port number where the sensor is connected, in the second, we specify the sensor type.

```

#define DHTPIN 15
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

```

In the *setup* function, we determine the speed of data exchange between the computer and ESP8266 through the virtual serial port, configure the GPIO2 port as an output for controlling the relay module, and start the DHT sensor. Next, we initialize the variables that were created in Arduino IoT Cloud (*thingProperties.h*), connected to the cloud via the Wi-Fi network (network connection parameters SSID, PASS, KEY are stored in the file *Secret.h*).

```
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  delay(1500);
  dht.begin();
  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}
```

In the *loop* function, we call the *ArduinoCloud.update()* function to update data in the Arduino IoT Cloud and the *DHT_SENSOR_READ()* function, which reads the temperature and humidity values from the sensor and outputs these data to the port monitor (to monitor the sensor's performance).

```
void loop() {
  ArduinoCloud.update();
  DHT_SENSOR_READ();
}
```

The remote control mechanism is implemented in the *onLedChange()* function. Depending on the state of the variable “led” (Switch widget), the relay module that controls ventilation, heating, air conditioning is turned on or off.

```
void onLedChange(){
  if (led==1)
    digitalWrite (2, HIGH);
  else
    digitalWrite (2, LOW);
}
```

The *DHT_SENSOR_READ()* function reads temperature and humidity values from the DHT11 combined sensor and calls the *AutoControl()* function, which performs automatic control mode depending on the given cooling/heating algorithm.

```
void DHT_SENSOR_READ(){
  float K_Filter = 3.0;
  float h, t, h_previous, h_current, t_previous, t_current;
  float dewpoint;
  h = dht.readHumidity();
  h_previous = h_current;
  h_current = h_previous + (h - h_previous)/ K_Filter;
  t = dht.readTemperature();
  t_previous = t_current;
  t_current = t_previous + (t - t_previous)/ K_Filter;
  temperature = t_current;
  humidity = h_current;
  dewpoint = dewPoint();AutoControl();
  Serial.print("Temperature - "); Serial.println(t);
  Serial.print("Humidity - "); Serial.println(h);
  delay(1000);
}
```

The *dewPoint* () function determines the value of the dew point by the values of temperature and humidity, the accuracy of which is increased by the weighted moving average method.

```
float dewPoint(){
    float a=17.27, float b=237.7;
    float funRhT = a* temperature/(b+ temperature);
    funRhT += log(humidity/100);
    float drew = b*funRhT/(a-funRhT);
    return drew;
}
```

The *AutoControl* () function implements algorithms for regulating automated ventilation and air conditioning systems based on the value of the dew point or basic thermodynamic parameters [13]. The code of this function can be adapted to any task of climate control according to the rules of fuzzy logic [14] and add automatic notifications using webhooks through the IFTTT service, for example, sending notifications to the Telegram channel or email.

```
void AutoContro(){
    if ((temperature > 22) || (humidity >65))
        digitalWrite (2, HIGH);
    else
        digitalWrite (2, LOW);
    delay (1500);
}
```

2.2.3 Programming an ESP8266 to transfer data to the Arduino cloud

Let's add the DHT.h library to the Arduino Cloud Editor. To do this, select Libraries in the vertical menu and enter DHT in the search query window. Among the proposed libraries, select DHT SENSOR LIBRARY FOR ESPX (Figure 18). We physically connected the NodeMCU module with the connected DHT11 sensor (Figure 6) and the LED simulating the operation of the relay (installed on the module by the manufacturer) to the computer. If the Arduino IoT Cloud does not automatically detect the connected module, you must install the Arduino Create Agent program. Figure 19 shows that the NodeMCU module is connected to COM9 and you can start verifying the project. If there are no errors, select the “Load” button and download the sketch to the NodeMCU module.

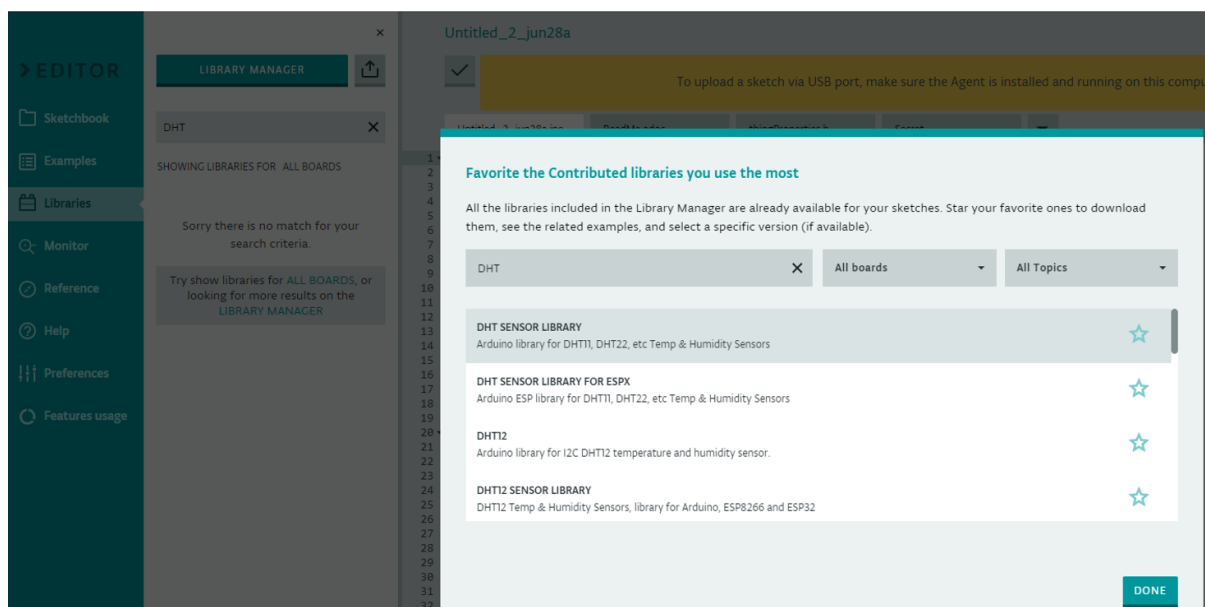


Figure 18: Adding the DHT library in the Arduino Cloud Editor

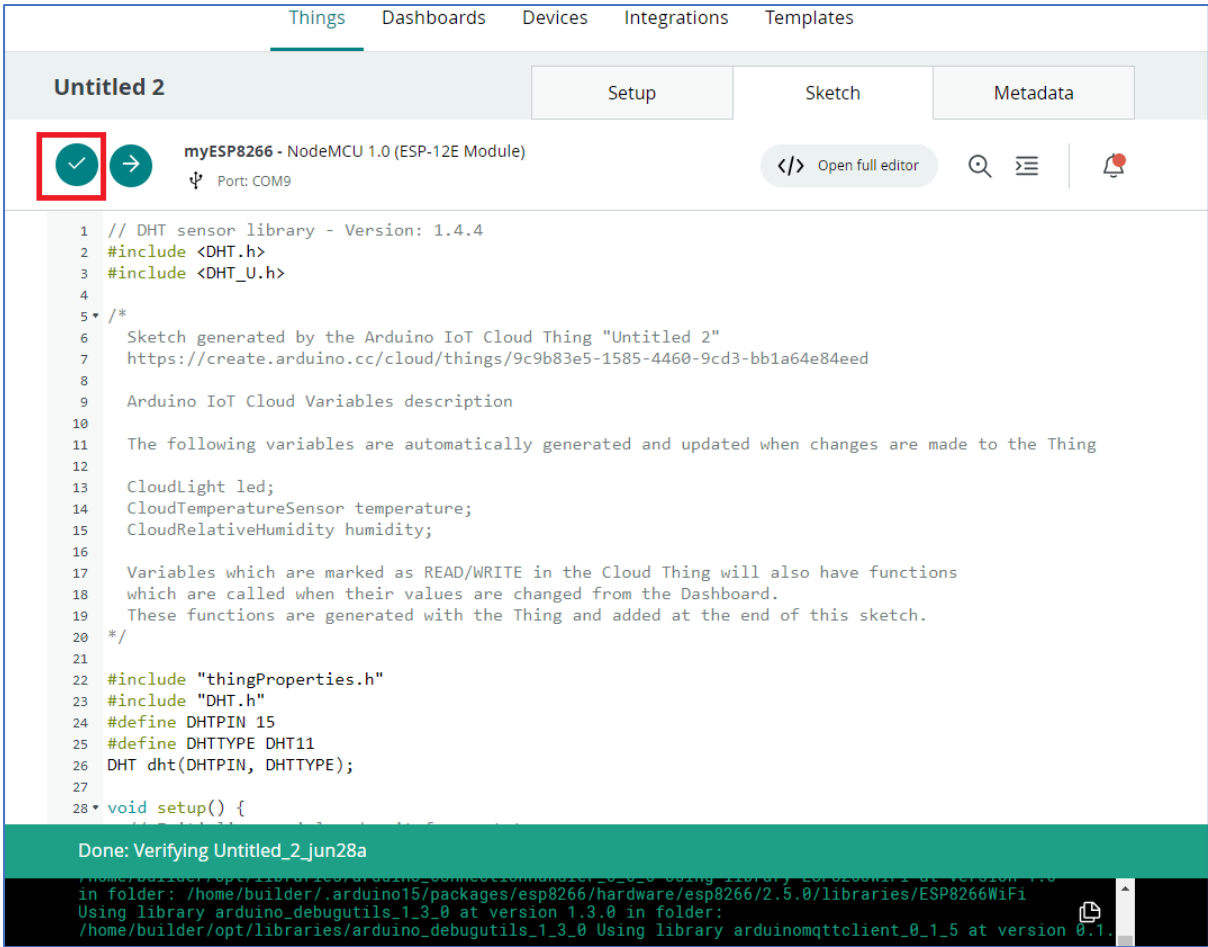


Figure 19: Checking code for errors in Arduino IoT Cloud

Open the “Serial Monitor” mode (Figure 20) and observe the temperature and humidity data measured by the sensor and transmitted by the module to the computer through the virtual serial port.

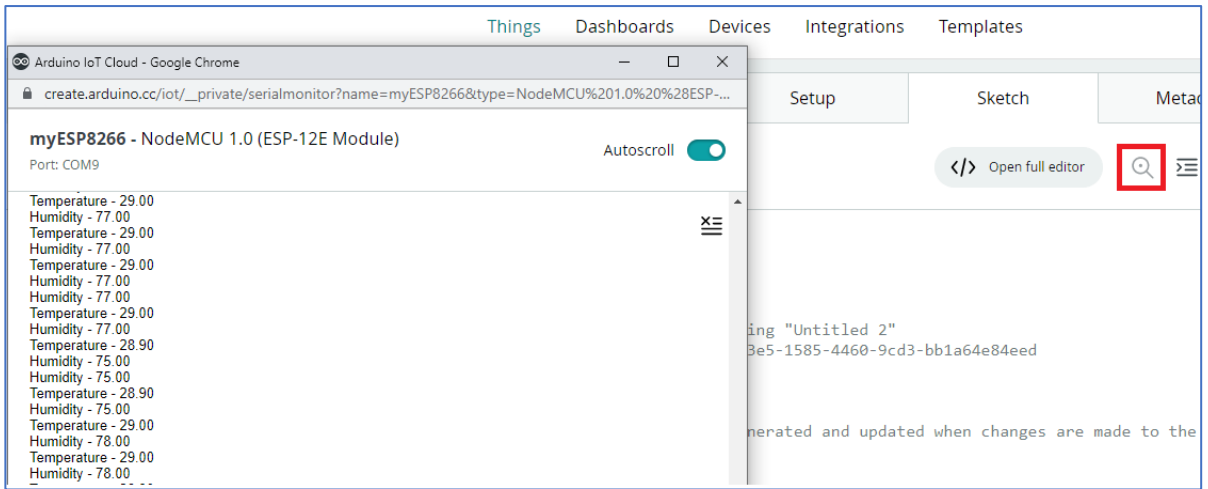


Figure 20: Output of NodeMCU module information to Arduino IoT Cloud in “Serial Monitor” mode

3. Results

The climate monitoring system is based on the ESP8266 microcontroller, which is part of the NodeMCU or WeMos D1 mini platform. The microcontroller organizes data transfer via Wi-Fi to the

Arduino IoT Cloud platform. A DHT11 sensor was used as a combined temperature and humidity sensor. The GPIO2 port is used to control the executive device in manual or automatic control mode, depending on the given cooling/heating algorithm. The experimental sample is implemented on the NodeMCU module, as an indication of the remote control mode, a blue LED built into the board is used, which is connected by the cathode to the GPIO2 port (Figure 21). This must be considered because some relay modules (Figure 6) use a p-n-p type transistor (switched on by a logic 0 level).

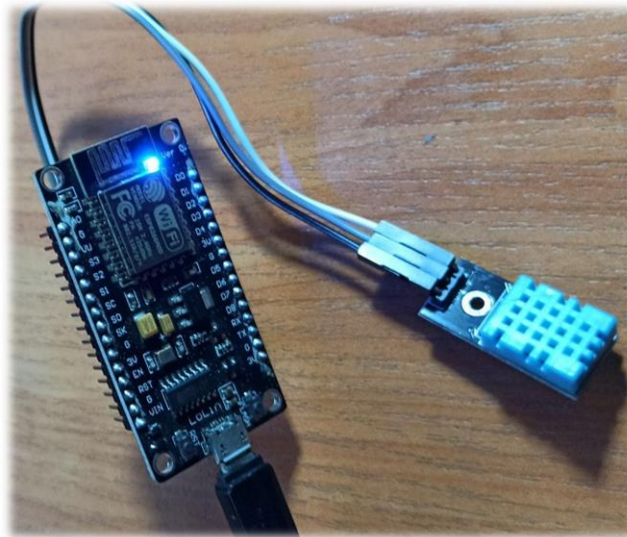


Figure 21: An experimental sample of the climate control system based on the NodeMCU module

The online monitoring and control panel that was created on the Arduino IoT Cloud platform is shown in Figure 22. For the convenience of working with the Arduino IoT Cloud dashboard, you can install the IoT Remote mobile application on your phone from the official Google Play or App Store (Figure 22).

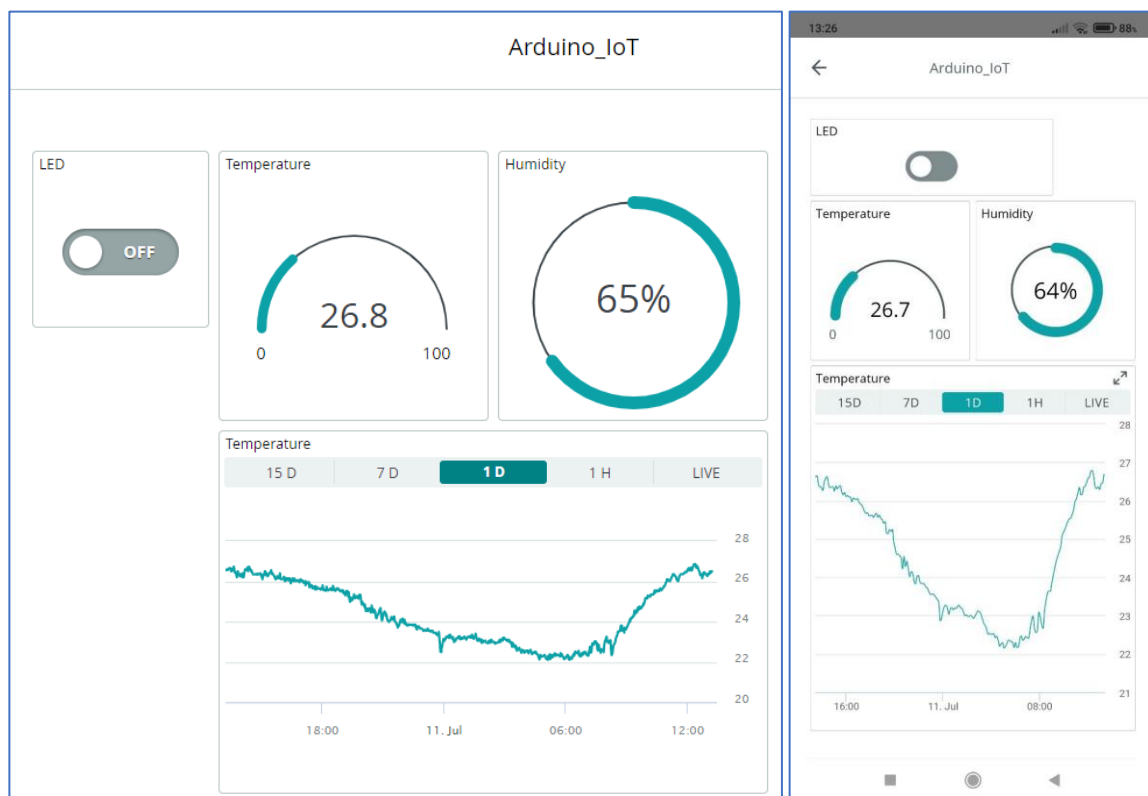


Figure 22: The dashboard of the climate monitoring system using Arduino IoT Cloud

During experimental research, the proposed system made it possible to apply the weighted moving average method to increase the accuracy of temperature and humidity measurements. At a temperature of 26.8 °C and a humidity of 65%, the calculated value of the dew point is 19.65 °C, which corresponds to the reference table values.

4. Conclusion

Control of storage temperature and air humidity is important for various technological processes. The use of wireless technologies provides some advantages: ease of installation and operation, scalability and simplicity of the solution, and integration of mobile devices. Wireless technology is an attractive choice for the renovation, repair, and installation of new systems. IoT-based temperature and humidity monitoring effectively reduce the cost of human resources for temperature and humidity control.

Arduino Cloud IoT allows the user to collect, analyze and manage data from a mobile application and web dashboard.

The article proposes to apply the weighted moving average method to increase the accuracy of measuring temperature and humidity parameters, which are used to calculate the value of the dew point or thermodynamic air parameters, which are necessary for the algorithms of automated ventilation and air conditioning systems.

The climate control system is implemented based on the IoT module NodeMCU (WeMos), the combined sensor DHT11, and the relay module with minimal cost and the possibility of adaptation to the monitoring and control system of the premises of the educational institution. Current climate data are available online. The operator, who directly controls the site, sees all the information in real-time on a digital panel. This, in turn, allows for efficient use of financial resources and, upon the improvement of this system, to automatically set the temperature on the premises, during working or non-working hours, and monitor the state of emergency.

5. References

- [1] V. Vychuzhanyn, S. Maliuta, Digital device for determining the thermodynamic parameters of air on the FPGA company Altera, Circuit design, 4, 2005, pp. 30-33.
- [2] S. Tsyrlunyk, Mobile applications and online WI-FI monitoring platforms of weather stations, Open educational e-environment of modern University, volume 9, 2020, pp. 181–192. doi: 10.28925/2414-0325.2020.9.15.
- [3] S. Tsyrlunyk, M. Tsyrlunyk, Temperature and humidity monitoring system on the IOT module, in: Proceedings of the II NPK. Computer Technologies of Data Processing (CTDP-2021), Vinnytsia, 2021, pp. 105-108. URL: <https://jktod.donnu.edu.ua/article/view/11664/11539>.
- [4] Arduino - Temperature Humidity Sensor, 2022. URL: <https://arduinogetstarted.com/tutorials/arduino-temperature-humidity-sensor>.
- [5] S. Tsyrlunyk, V. Tromsyuk, M. Tsyrlunyk, P. Rymar, Energy Monitoring System based on IoT, in: CEUR Workshop Proceedings (CEUR-WS. org). 2021. Vol. 3039. P. 136–153.
- [6] A. Javed, Building Arduino Projects for the Internet of Things, Experiments with Real-World Applications. United States of America: Apress Media, LLC, 2016, pp. 15-34.
- [7] K. Küçük, C. Bayılmış, D. L. Msongaleli, Designing real-time IoT system course: prototyping with cloud platforms, laboratory experiments and term project, The International Journal of Electrical Engineering & Education, 58(3), 2021, pp. 743-772. doi:10.1177/0020720919862496.
- [8] V. Kucheruk, I. Kolomiichuk, The use of recursive filters to reduce random measurement errors, Bulletin of the Engineering Academy of Ukraine, 1, 2013, pp. 251-254.
- [9] Arduino IoT Cloud, 2022. URL: <https://create.arduino.cc/iot>.
- [10] Kurniawan, Arduino IoT Cloud, in: Beginning Arduino Nano 33 IoT. Apress, Berkeley, CA, 2021. doi: 10.1007/978-1-4842-6446-1_5.
- [11] Hao Xu, Chul-Won Kim. "Design and implementation of LED lighting control system using Arduino Yun and cloud in IoT". The Journal of the Korea institute of electronic communication sciences 11.10 (2016): 983-988.

- [12] Getting Started With the Arduino IoT Cloud, 2022. URL: <https://docs.arduino.cc/arduino-cloud/getting-started/iot-cloud-getting-started>.
- [13] D. Karabekova, P. Kissabekova, V. Kucheruk, E. Mussenova, & S. Azatbek. Main characteristics of the heat flow meter, *Eurasian Physical Technical Journal*, 19.2(40), 2022, pp. 71-74. doi: 10.31489/2022No2/71-74.
- [14] I. Olenych, Smart home climate control system based on fuzzy logic controller, *Electronics and Information Technologies*, 17, 2022, pp. 26-35. doi: <https://doi.org/10.30970/eli.17>.