

УДК 004.3

DOI <https://doi.org/10.32838/2663-5941/2022.1/25>

Новак Д.С.

Київський національний університет технологій та дизайну

Мошенський А.О.

Національний університет харчових технологій

Лісовець С.М.

Таврійський національний університет імені В.І. Вернадського

Гуйда О.Г.

Таврійський національний університет імені В.І. Вернадського

Павленко Є.Є.

Київський національний університет технологій та дизайну

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО ОЦІНЮВАННЯ ТЕМПЕРАТУРИ, ВІДНОСНОЇ ВОЛОГОСТІ Й АТМОСФЕРНОГО ТИСКУ

Вимірюванням називають сукупність операцій, виконуваних для визначення кількісного значення величини. Вимірювання виконуються в усіх сферах людської діяльності. Зокрема, широко поширеними для вимірів величинами є параметри довкілля.

Необхідність отримання інформації про значення таких фізичних величин, як температура, відносна вологість та атмосферний тиск, виникає при контролі умов праці, умов виробництва та зберігання продукції у приміщеннях, при здійсненні заходів екологічного моніторингу, у метеорологічних спостереженнях і прогнозах, для забезпечення єдності вимірювань при проведенні повірки та/або калібрування засобів вимірювань тощо. Для зниження трудових і тимчасових витрат на організацію вимірювальних процедур може бути використана система, що дозволяє автоматизувати процес збору, обробки та збереження вимірювальних даних. Така система має бути мобільною, мати можливості нарощування елементної бази та заміни сенсорів фізичних величин для забезпечення універсальності застосування.

Метою роботи є розроблення автоматизованої бездротової системи вимірювань температури, відносної вологості й атмосферного тиску навколишнього середовища. Для досягнення поставленої мети необхідне виконання таких задач: формування вимог до системи вимірювань, розроблення структури системи вимірювань, вибору елементної бази, розроблення програмного забезпечення.

У процесі дослідження проводилося: вивчення сфер застосування системи вимірювань температури, відносної вологості й атмосферного тиску навколишнього середовища, формування вимог до системи вимірювань, вибір вимірюваних величин, розроблення структури та програмного забезпечення системи.

Досліджено та проаналізовано мікроконтролери та сенсори, розроблено серверну частину. Програмування мікроконтролеру відбувалося з використанням мови програмування C++. Розроблено апаратно-програмний комплекс, що складається із серверної частини на базі програмного забезпечення, написаного мовою програмування PHP, та реляційної бази даних MySQL, а також розроблено клієнт-серверну частину для сенсорів на базі мікроконтролера ESP32.

Ключові слова: мікроконтролер, сенсор, сервер, ESP32, BME280, PHP, MySQL.

Постановка проблеми. Вимірювання параметрів довкілля зачіпає багато сфер людської діяльності, наприклад, промисловість або лабораторію високоточних вимірювань, приватну садибу або метеорологічну станцію. Також для якісного виробництва будь-якої продукції необхідні певні умови навколишнього середовища. Саме тому останнім часом роблять безперервні вимірювання параметрів довкілля та їх автоматизацію.

Система вимірювань параметрів довкілля (далі – Система) може застосовуватися у різних галузях, таких як екологічний моніторинг, система «Розумний дім», реєстрація факторів при перевірці засобів вимірювань (ЗВ), а також вимірювання параметрів мікроклімату виробничих і житлових приміщень.

На підставі вимог, що пред'являються до системи при вимірюванні параметрів навколишнього

середовища у конкретних галузях застосування, був сформований список фізичних величин, які вимірюватимуться системою.

У випадку контролю параметрів навколишнього середовища включає вимір таких величин, як: температура навколишнього середовища, відносна вологість навколишнього середовища й атмосферний тиск.

Постановка задачі. Метою роботи є розроблення автоматизованої бездротової системи вимірювань параметрів довкілля. У процесі дослідження проводилися: вивчення сфер застосування системи вимірювань параметрів довкілля, формування вимог до системи вимірювань, вибір вимірюваних величин, розроблення структури та програмного забезпечення системи.

Внаслідок дослідження створено автоматизовану бездротову систему вимірювань параметрів навколишнього середовища, яка вимірює температуру, відносну вологість навколишнього середовища й атмосферний тиск.

ESP8266 (рис. 1) та ESP32 (рис. 2) – це недорогі модулі Wi-Fi, котрі ідеально підходять для самостійних проєктів у галузі інтернету (IoT). ESP32 є наступником ESP8266. Він має більше ядер процесора, більш швидкий Wi-Fi, більше GPIO і підтримує Bluetooth версії 4.2. Крім того, ESP32 поставляється із сенсором дотику, вбудованим датчиком Хола та датчиком температури. Обидві плати недорогі, але ESP32 більш дорожча. Обидва чіпи мають 32-бітний процесор. ESP32 – двоядерний процесор із частотою від 160 до 240 МГц, а ESP8266 – одноядерний, працює на частоті 80 МГц. Мікроконтролер RTL8710 (рис. 3) має такі основні переваги: широкий спектр сумісних пристроїв, можливість роботи навіть за екстремальних температур. Також він майже повністю відповідає по виводах модулю ESP8266, однак RTL8710 дорожче за ESP8266 та ESP32. У нашому випадку кращим вибором буде ESP32.

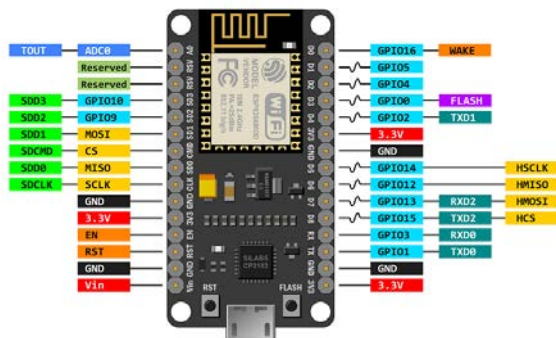


Рис. 1. Мікроконтролер ESP8266

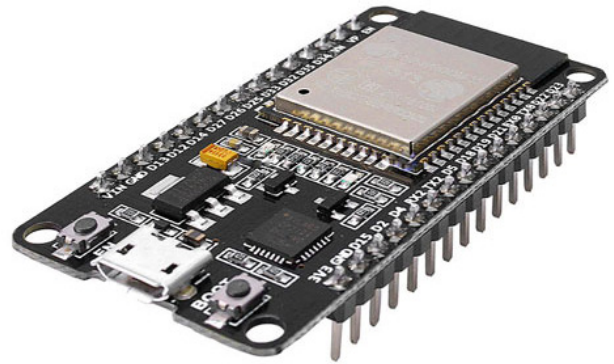


Рис. 2. Мікроконтролер ESP32

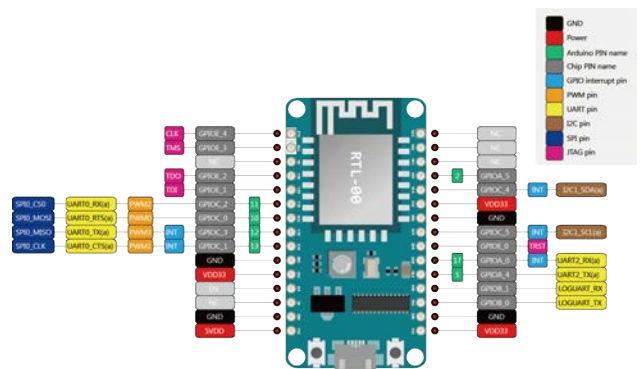


Рис. 3. Мікроконтролер RTL8710

Сенсор DHT22 (рис. 4) має заводське калібрування і характеризується низьким енергоспоживанням. Сенсор BME680 (рис. 5) має надлишкові характеристики та високу ціну. Сенсор DS18B20 (рис. 6) має серйозний недолік (зняття даних гальмує виконання подальших команд). BMP280 (рис. 7) є покращеною версією датчика BMP180 (рис. 8) і відрізняється від нього меншими розмірами, зниженим енергоспоживанням, високою точністю роботи та наявністю точного заводського калібрування і двома послідовними інтерфейсами: I2C і SPI. Датчик BMP280, створений спеціально для додатків, де потрібні малі розміри та знижене споживання енергії, тому обрати саме цей датчик буде найбільш доцільно.

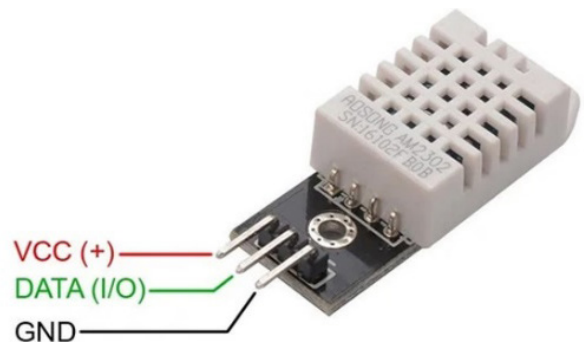


Рис. 4. Сенсор DHT22

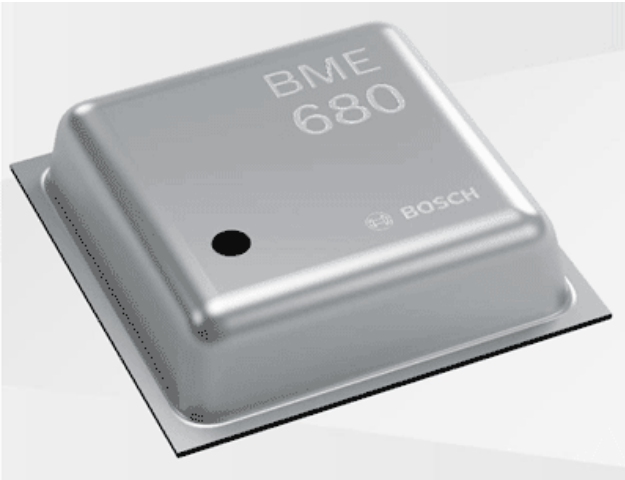


Рис. 5. Сенсор BME680

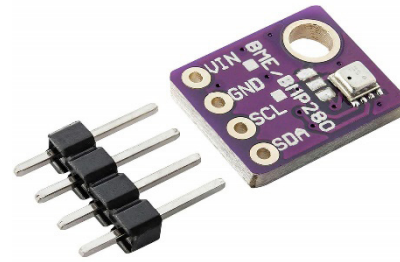


Рис. 7. Сенсор BMP280

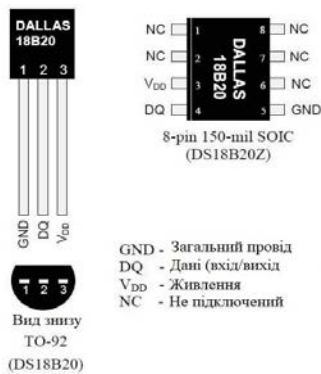


Рис. 6. Сенсор DS18B20

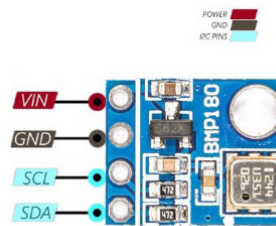


Рис. 8. Сенсор BME180

Для роботи нам знадобиться (рис. 9):
 – Плата ESP32 та середовище розробки Arduino.
 – Зареєстрований домен на хостингу.

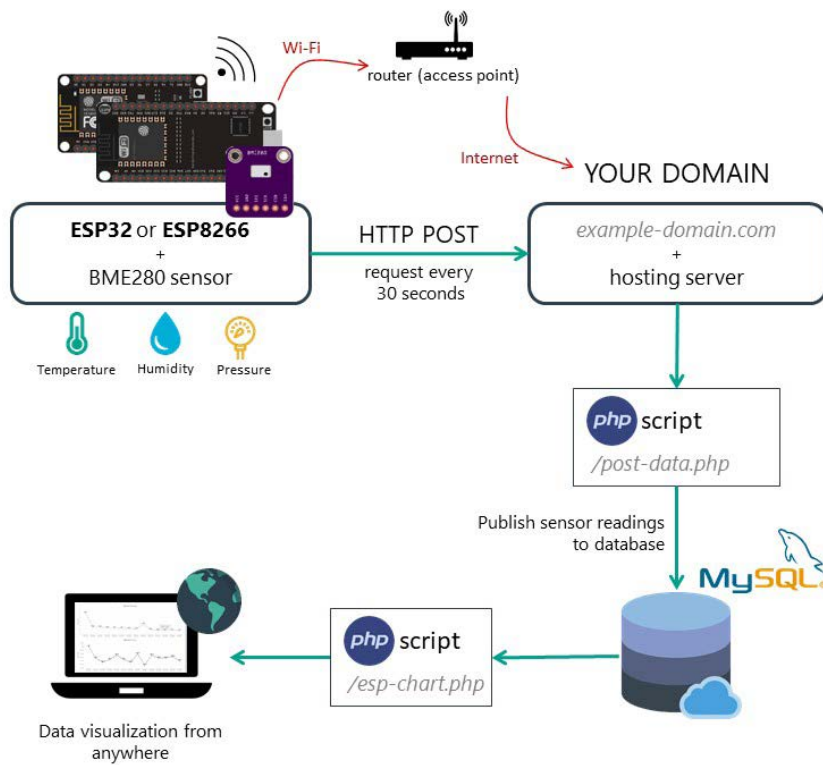


Рис. 9. Графічна ілюстрація роботи

- Скрипт PHP для запису даних у базу даних (БД) та відображення їх на веб-сторінці.
- База даних для зберігання показань.
- Скрипт PHP для графічного формування результатів досліджень.

Реєструємо домен для доступу до нашого додатку (будемо використовувати хостинг Bluehost).

Після реєстрації облікового запису та вибору доменного імені заходимо у консоль. Після цього відбувається створення БД, логіну, пароля та таблиці SQL. Переходимо до панелі керування (рис. 10) і шукаємо «phpMyAdmin».

У лівому меню вибираємо нашу базу та відкриваємо вкладку «SQL». Далі вводимо наступні рядки коду для створення таблиці (рис. 11), натис-

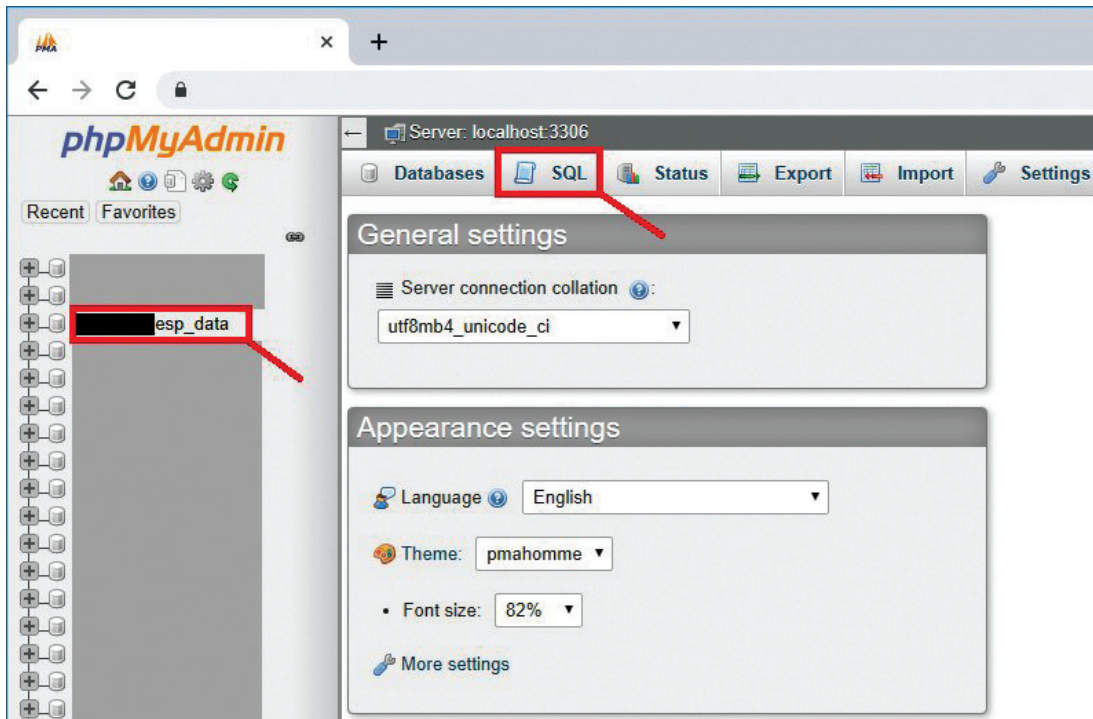


Рис. 10. phpMyAdmin

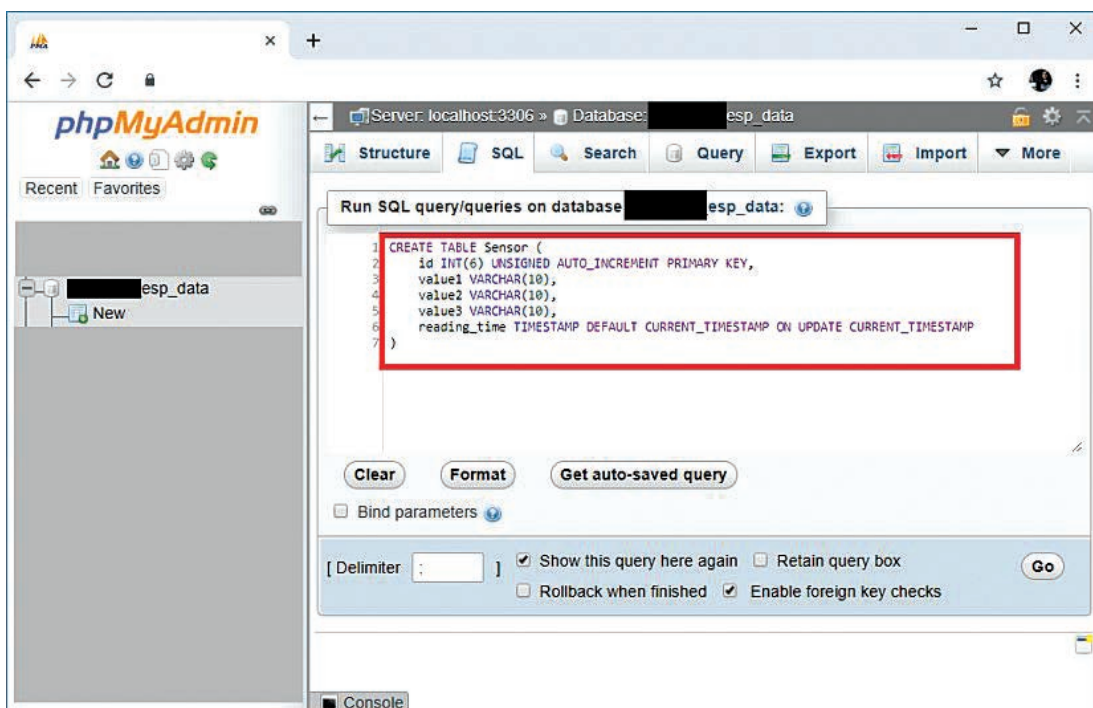


Рис. 11. Створення таблиці в phpMyAdmin

каємо кнопку «Go» та бачимо таблицю під назвою Sensor (рис. 12) у нашій базі даних.

В опції cPanel знаходимо «File Manager» (рис. 13) і вибираємо опцію public_html і натискаємо «+ File».

Створюємо новий файл post-data.php у папці /public_html пишемо туди рядки коду (рис. 14).

Створимо ще один файл (рис. 15) у папці /public_html. Він графічно відобразить дані на веб-сторінці (назвемо його esp-chart.php).

Для графічного відображення даних ми будемо використовувати бібліотеку Highcharts. Створимо три – залежності температури, вологості та тиску залежно від часу. На залежностях відображається максимум 40 пікселів, показання додаються кожні 30 секунд.

Ми використовуємо Arduino IDE, тому потрібно, щоб плати були встановлені. Також потрібні бібліотеки BME280 та Adafruit_sensor.

Для того, щоб код для мікроконтролеру ESP32 (рис. 16) працював, необхідно вказати свої дані мережі (SSID і пароль), також потрібно ввести доменне ім'я.

Якщо відкрити веб-сторінку (рис. 17) за URL-адресою: <http://example.com/esp-chart.php>, то будуть відображені залежності температури, відносної вологості й атмосферного тиску від часу. Для оновлення даних потрібно перезавантажити веб-сторінку. Також є можливість проведення модифікації отриманих даних у phpMyAdmin (рис. 18).

Висновки. Під час виконання роботи проведено аналіз предметної сфери, актуальності розроблення системи дистанційного оцінювання параметрів навколишнього середовища, визначено головні вимоги до системи та її функціоналу.

Розроблено апаратно-програмний комплекс, що складається із серверної частини на базі програмного забезпечення (PHP + MySQL) і клієнтської частини на базі мікроконтролера ESP32 та сенсора BME 280.

Виконано експериментальні дослідження зміни показників температури, відносної вологості й атмосферного тиску від часу. Проведені експерименти показали ефективність застосування мікроконтролера ESP32 та сенсора BME 280 для оцінювання параметрів навколишнього середовища.

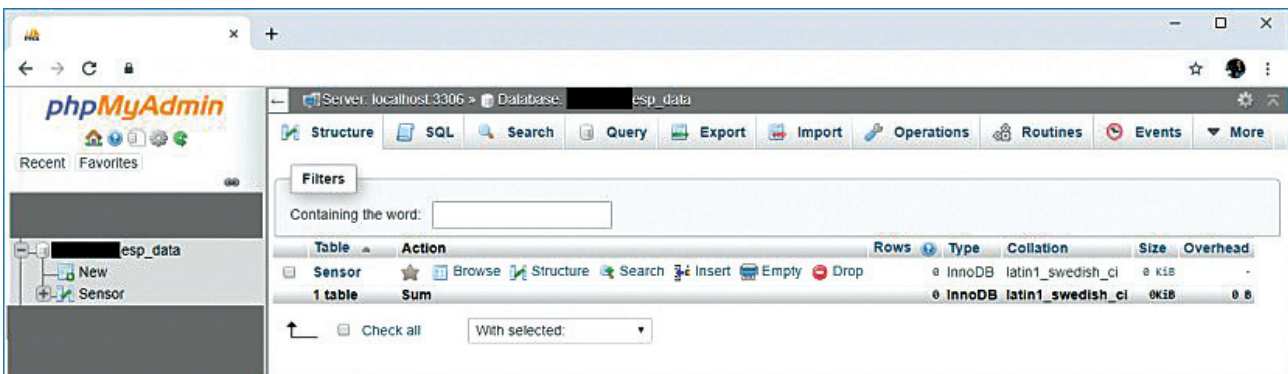


Рис. 12. Таблиця «Sensor» у базі даних

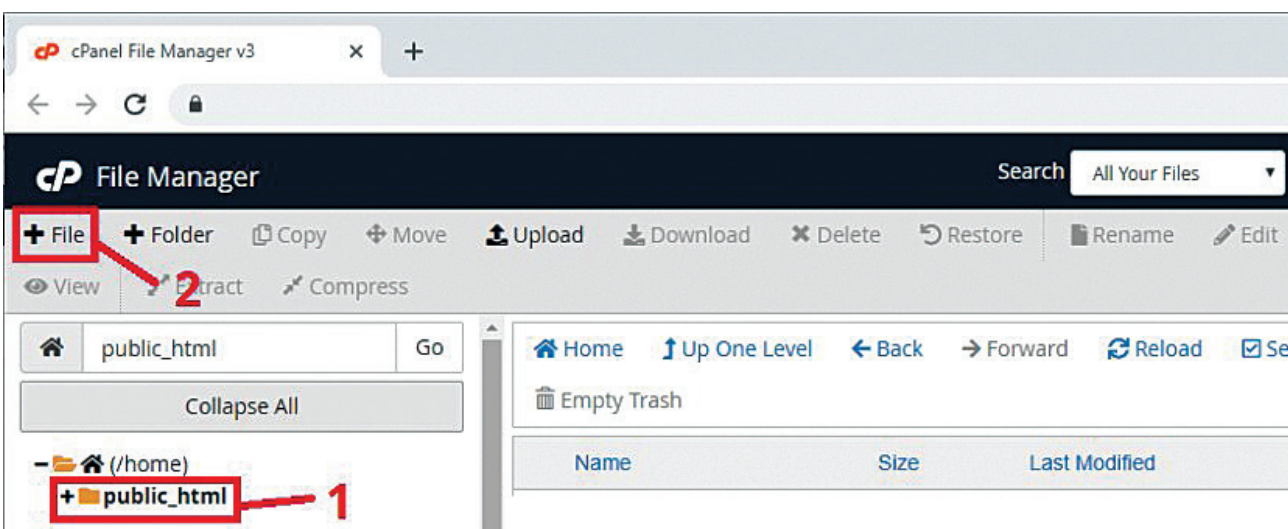


Рис. 13. Файловий менеджер

```

<?php
$servername = 'localhost';
// Тут вказуємо назву БД
$dbname = 'ВКАЖІТЬ_НАЗВУ_БД';
// Вказуємо ім'я користувача
$username = 'ВКАЖІТЬ_ІМ'Я_КОРИСТУВАЧА';
// Вказуємо пароль
$password = 'Вкажіть_пароль';
// Рекомендується не змінювати цей API-ключ, він повинен збігатися з ключем у скетчі для плати
$api_key_value = 'tPmAT5Ab3j7F9';
$api_key = $value1 = $value2 = $value3 = '';
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $api_key = test_input($_POST['api_key']);
    if($api_key == $api_key_value) {
        $value1 = test_input($_POST['value1']);
        $value2 = test_input($_POST['value2']);
        $value3 = test_input($_POST['value3']);
        // Створюємо з'єднання
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Перевіряємо з'єднання
        if ($conn->connect_error) {
            die('Connection failed: ' . $conn->connect_error);
        }
        $sql = 'INSERT INTO Sensor (value1, value2, value3)
VALUES (' . $value1 . ', ' . $value2 . ', ' . $value3 . ')';

        if ($conn->query($sql) == TRUE) {
            echo "New record created successfully";
        }
        else {
            echo "Error:". $sql. "<br>" . $conn->error;
        }
        $conn->close();
    }
    else {
        echo "Wrong API Key provided.";
    }
}
else {
    echo "No data posted with HTTP POST."
}
function test_input($data) {
    $data = trim ($ data);
    $data = stripslashes ($ data);
    $data = htmlspecialchars ($ data);
    return $data;
}
Перед збереженням файла введіть дані в наступні рядки:
// Тут вказуємо назву БД
$dbname = 'ВКАЖІТЬ_НАЗВУ_БД';
// Вказуємо ім'я користувача
$username = 'ВКАЖІТЬ_ІМ'Я_КОРИСТУВАЧА';
// Вказуємо пароль
$password = 'Вкажіть_пароль';
?>

```

Рис. 14. PHP код

Список літератури:

1. Cameron, N. Electronics Projects with the ESP8266 and ESP32. Apress, 2021.
2. URL: <https://link.springer.com/book/10.1007/978-1-4842-6336-5>.
3. Pulver, T. Hands-On Internet of Things with MQTT: Build connected IoT devices with Arduino and MQ Telemetry Transport (MQTT). Packt Publishing Ltd, 2019.
4. Spanulescu, S. Esp32 Programming for the Internet of Things. Lulu Press, Inc, 2018.
5. Kurniawan, A. Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32. Packt Publishing Ltd, 2019.
6. Butler, T. PHP & MySQL: Novice to Ninja. SitePoint, 2022.

Novak D.S., Moshenskyi A.O., Lisovets S.M., Guida O.G., Pavlenko Ye.Ye. INFORMATION SYSTEM FOR REMOTE EVALUATION OF TEMPERATURE, RELATIVE HUMIDITY AND ATMOSPHERIC PRESSURE

A measurement is a set of operations performed to determine the quantitative value of a quantity. Measurements are made in all spheres of human activity. In particular, environmental parameters are widely used for measurements.

```

<?php
$servername = 'localhost';
// Тут вказуємо адресу БД
$dbname = 'ВКЛАДІТЬ_НАЗВУ_БД';
// Вказуємо ім'я користувача
$username = 'ВКЛАДІТЬ_ІМ'Я_КОРИСТУВАЧА';
// Вказуємо пароль
$password = 'ВКЛАДІТЬ_ПАРОЛЬ';
// Створюємо з'єднання
$conn = new mysqli($servername, $username, $password, $dbname);
// Перевіряємо з'єднання
if ($conn->connect_error) {
die('Connection failed: ' . $conn->connect_error);
}
$sql = 'SELECT id, value1, value2, value3, reading_time FROM Sensor order by reading_time desc limit 40';
$result = $conn->query($sql);
while ($data = $result->fetch_assoc()){
$sensor_data[] = $data;
}
$readings_time = array_column($sensor_data, 'reading_time');
// час - UTC-2, замикаємо 0
*$i = 0;
foreach ($readings_time as $reading){
// Для зони UTC 1
$readings_time[$i] = date('Y-m-d H:i:s', strtotime('$reading - 1 hours'));
// Для зони UTC 0
$readings_time[$i] = date('Y-m-d H:i:s', strtotime('$reading + 4 hours'));
$i += 1;
}
$value1 = json_encode(array_reverse(array_column($sensor_data, 'value1')), JSON_NUMERIC_CHECK);
$value2 = json_encode(array_reverse(array_column($sensor_data, 'value2')), JSON_NUMERIC_CHECK);
$value3 = json_encode(array_reverse(array_column($sensor_data, 'value3')), JSON_NUMERIC_CHECK);
$reading_time = json_encode(array_reverse($readings_time), JSON_NUMERIC_CHECK);
/*echo $value1;
echo $value2;
echo $value3;
echo $reading_time;*/
$result->free();
$conn->close();
?>
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="https://code.highcharts.com/highcharts.js"></script>
<style>
body {
min-width: 310px;
max-width: 1280px;
height: 500px;
margin: 0 auto;
}
h2 {
font-family: Arial;
font-size: 2.5rem;
text-align: center;
}
</style>
<body>
<h2>ESP Weather Station</h2>
<div id="chart-temperature" class="container"></div>
<div id="chart-humidity" class="container"></div>
<div id="chart-pressure" class="container"></div>
<script>
var value1 = <?php echo $value1; ?>;
var value2 = <?php echo $value2; ?>;
var value3 = <?php echo $value3; ?>;

var reading_time = <?php echo $reading_time; ?>;
var chartT = new Highcharts.Chart({
chart: { renderTo: 'chart-temperature' },
title: { text: 'BME280 Temperature' },
series: [{
showInLegend: false,
data: value1
}],
plotOptions: {
line: { animation: false,
dataLabels: { enabled: true }
}
},
series: { color: '#059e8a' }
},
xAxis: {
type: 'datetime',
categories: reading_time
},
yAxis: {
title: { text: 'Temperature (Celsius)' },
//title: { text: 'Temperature (Fahrenheit)' }
},
credits: { enabled: false }
});
var chartH = new Highcharts.Chart({
chart: { renderTo: 'chart-humidity' },
title: { text: 'BME280 Humidity' },
series: [{
showInLegend: false,
data: value2
}],
plotOptions: {
line: { animation: false,
dataLabels: { enabled: true }
}
},
xAxis: {
type: 'datetime',
//dateTimeLabelFormats: { second: '%H:%M:%S' },
categories: reading_time
},
yAxis: {
title: { text: 'Humidity (%)' }
},
credits: { enabled: false }
});
var chartP = new Highcharts.Chart({
chart: { renderTo: 'chart-pressure' },
title: { text: 'BME280 Pressure' },
series: [{
showInLegend: false,
data: value3
}],
plotOptions: {
line: { animation: false,
dataLabels: { enabled: true }
}
},
series: { color: '#18009c' }
},
xAxis: {
type: 'datetime',
categories: reading_time
},
yAxis: {
title: { text: 'Pressure (hPa)' }
},
credits: { enabled: false }
});
</script>
</body>
</html>

```

Рис. 15. Код веб-додатку


```

#ifdef ESP32
#include <WiFi.h>
#include <HTTPClient.h>
#else
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#endif
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
//Вкажіть свої облікові дані мережі
const char* ssid = "ВКАЖІТЬ_СВІЙ_SSID";
const char* password = "ВКАЖІТЬ_ПАРОЛЬ";
//Вкажіть свій домен і шлях до файлу
const char* serverName = "http://example.com/post-data.php";
//Залиште ключ як є, якщо ви його зміните, то код не буде робити
//Тому, що API повинен співпадати з ключом в файлі PHP
String apiKeyValue = "tPmAT5Ab3j7F9";
Adafruit_BME280 bme; // I2C
//Adafruit_BME280 bme(BME_CS); // hardware SPI
//Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); // software SPI
void setup() {
Serial.begin(115200);
WiFi.begin(ssid, password);
Serial.println("Connecting");
while(WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
//Ви також можете використовувати бібліотеку Wire
bool status = bme.begin(0x76);
if (!status) {
Serial.println("Could not find a valid BME280 sensor, check wiring or change I2C address!");
while (1);
}
void loop() {
//Перевіряємо з'єднання WiFi
if(WiFi.status()== WL_CONNECTED){
HTTPClient http;
//Вкажіть URL або IP вашого домену
http.begin(serverName);
//Вказуємо тип даних
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
//Готуємо запит
String httpRequestData = "api_key=" + apiKeyValue + "&value1=" + String(bme.readTemperature()) +
"&value2=" + String(bme.readHumidity()) + "&value3=" + String(bme.readPressure()/100.0F) + "&";
Serial.print("httpRequestData: ");
Serial.println(httpRequestData);
//Для тесту ви можете закоментувати змінну httpRequestData вище
//і розкоментувати, то нижче (датчик BME280 не використовується)
//String httpRequestData = "api_key=tPmAT5Ab3j7F9&value1=24.75&value2=49.54&value3=1005.14";
//Відправляємо запит
int httpResponseCode = http.POST(httpRequestData);
//якщо ви робите запит з типом text/plain, розкоментуйте наступні рядки
//http.addHeader("Content-Type", "text/plain");
//int httpResponseCode = http.POST("Hello, World!");
//якщо ви робите запит з типом application/json, розкоментуйте наступні рядки
//http.addHeader("Content-Type", "application/json");
//int httpResponseCode = http.POST("{\"value1\": \"19\", \"value2\": \"67\", \"value3\": \"78\"}");
if (httpResponseCode>0) {
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
}
else {
Serial.print("Error code: ");
Serial.println(httpResponseCode);
}
//Звільняємо пам'ять
http.end();
}
else {
Serial.println("WiFi Disconnected");
}
//Відправляємо запит кожні 30 секунд
delay(30000);
}
}

```

Рис. 16. Код для мікроконтролеру ESP32

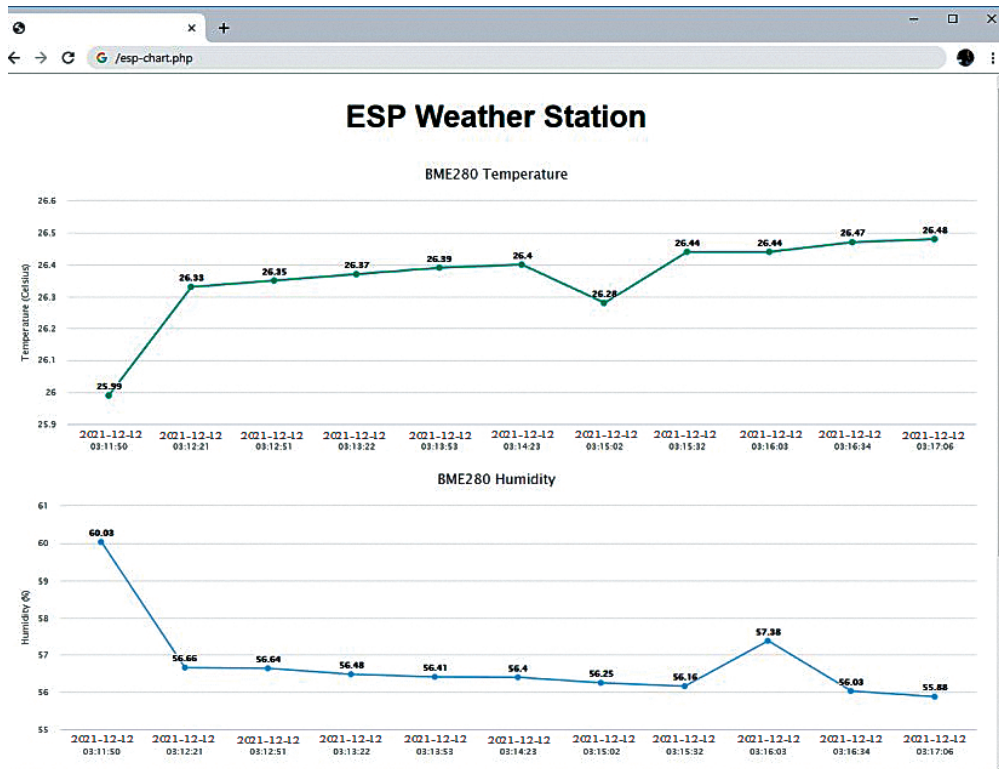


Рис. 17. Залежності температури, відносної вологості й атмосферного тиску від часу

				id	value1	value2	value3	reading_time
<input type="checkbox"/>				41	26.43	55.36	1002.88	2021-12-12 08:26:17
<input type="checkbox"/>				42	26.92	53.93	1003.01	2021-12-12 08:26:48
<input type="checkbox"/>				43	27.08	55.20	1002.94	2021-12-12 08:27:18
<input type="checkbox"/>				44	27.15	55.32	1002.99	2021-12-12 08:27:49
<input type="checkbox"/>				45	27.17	53.69	1002.93	2021-12-12 08:28:20
<input type="checkbox"/>				46	27.19	53.70	1002.95	2021-12-12 08:28:50
<input type="checkbox"/>				47	27.22	53.20	1002.96	2021-12-12 08:29:21
<input type="checkbox"/>				48	27.26	52.60	1002.97	2021-12-12 08:29:52
<input type="checkbox"/>				49	27.31	52.45	1002.99	2021-12-12 08:30:22
<input type="checkbox"/>				50	27.34	52.41	1002.97	2021-12-12 08:30:53
<input type="checkbox"/>				51	27.37	52.20	1002.90	2021-12-12 08:31:24
<input type="checkbox"/>				52	27.42	52.25	1002.96	2021-12-12 08:31:54
<input type="checkbox"/>				53	27.45	54.22	1002.99	2021-12-12 08:32:25
<input type="checkbox"/>				54	27.47	52.30	1002.95	2021-12-12 08:32:56
<input type="checkbox"/>				55	27.46	51.74	1002.95	2021-12-12 08:33:27
<input type="checkbox"/>				56	27.53	52.11	1002.93	2021-12-12 08:33:58
<input type="checkbox"/>				57	27.53	52.07	1003.00	2021-12-12 08:34:28
<input type="checkbox"/>				58	27.57	52.12	1002.96	2021-12-12 08:35:00

Рис. 18. Відображення бази даних у phpMyAdmin

The need to obtain information about the value of such physical quantities as temperature, relative humidity and atmospheric pressure arises when monitoring working conditions, conditions for production and storage of products in premises, when implementing environmental monitoring measures, in meteorological observations and forecasts to ensure the uniformity of measurements during verification and/or calibration of measuring instruments, etc. To reduce labor and time costs for the organization of measurement procedures, a system can be used to automate the process of collecting, processing and storing measurement data. Such a system should be mobile, have an increase in the element base and replacement of sensors of physical quantities to ensure universal application.

The purpose of this work is to develop an automated wireless system for measuring the temperature, relative humidity and atmospheric pressure of the environment. To achieve the set goals, it is necessary to perform the following tasks: formation of requirements for the measurement system, development of the structure of the measurement system, selection of the element base, software development.

In the process of research, the following was carried out: a study of the areas of application of the system for measuring temperature, relative humidity and atmospheric pressure of the environment, the formation of requirements for the measurement system, the choice of measured quantities, the development of the structure and software of the system.

Microcontrollers and sensors have been researched and analyzed; the server part has been developed. The microcontroller was programmed using the C++ programming language. A hardware-software complex has been developed, consisting of a server part based on software written in the PHP programming language and a MySQL relational database, as well as a client-server part for sensors based on the ESP32 microcontroller.

Key words: *microcontroller, sensor, server, ESP32, BME280, PHP, MySQL.*