

ДОДАТОК А

Програмний код розроблюваного ПЗ

Лістинг А.1 – Програмний код роботи веб-сервера

```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
const char* ssid = "your_ssid";
const char* password = "your_password";
AsyncWebServer server(80);
void setup() {
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
    Serial.println(WiFi.localIP());
    server.on("/", HTTP_GET, handleRoot);
    server.on("/sensor_data", HTTP_GET, handleSensorData);
    server.on("/camera", HTTP_GET, handleCameraStream);
    server.on("/open_valve", HTTP_POST, handleOpenValve);
    server.on("/close_valve", HTTP_POST, handleCloseValve);
    server.on("/start_washing", HTTP_POST, handleStartWashing);
    server.begin();
}
```

Лістинг А.2 – Функція SendHTML() для створення HTML-сторінки для веб-інтерфейсу

```
String SendHTML() {
```

```

String ptr = "<!DOCTYPE html> <html>\n";
ptr += "<meta charset=\"utf-8\"><head><meta name=\"viewport\"
content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
ptr += "<title>Monitoring System Control</title>\n";
ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;} #video-container { width: 100%; height: auto;}";
ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}
h3 {color: #444444;margin-bottom: 50px;} img{height: auto; width: 80%; max-width:
640px;}\n";
ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
ptr += "select {margin: 10px; padding: 5px;}\n";
ptr += "button {margin: 10px; padding: 10px 20px;}\n";
ptr += "</style>\n";
ptr += "<script>\n";
ptr += "var intervalId;\n";
ptr += "function fetchData() {\n";
ptr += "  var xhr = new XMLHttpRequest();\n";
ptr += "  xhr.onreadystatechange = function() {\n";
ptr += "    if (this.readyState == 4 && this.status == 200) {\n";
ptr += "      var data = JSON.parse(this.responseText);\n";
ptr += "      document.getElementById('temp').innerText = data.temperature;\n";
ptr += "      document.getElementById('tempState').innerText =
data.tempState;\n";
ptr += "      document.getElementById('waterLevel').innerText =
data.distance;\n";
ptr += "      document.getElementById('waterState').innerText =
data.waterState;\n";
ptr += "      document.getElementById('valveState').innerText =
data.valveState;\n";

```

```

ptr += "    }\n";
ptr += "  };\n";
ptr += "  xhr.open('GET', '/sensor_data', true);\n";
ptr += "  xhr.send();\n";
ptr += "}\n";
ptr += "function updateVideo() {\n";
ptr += "    document.getElementById('video').src = '/camera?rand=' +
Math.random();\n";
ptr += "}\n";
ptr += "function openValve() {\n";
ptr += "    var xhr = new XMLHttpRequest();\n";
ptr += "    xhr.open('POST', '/open_valve', true);\n";
ptr += "    xhr.onreadystatechange = function() {\n";
ptr += "        if (this.readyState == 4 && this.status == 200) {\n";
ptr += "            document.getElementById('status').innerText = this.responseText;\n";
ptr += "        }\n";
ptr += "    };\n";
ptr += "    xhr.send();\n";
ptr += "}\n";
ptr += "function closeValve() {\n";
ptr += "    var xhr = new XMLHttpRequest();\n";
ptr += "    xhr.open('POST', '/close_valve', true);\n";
ptr += "    xhr.onreadystatechange = function() {\n";
ptr += "        if (this.readyState == 4 && this.status == 200) {\n";
ptr += "            document.getElementById('status').innerText = this.responseText;\n";
ptr += "        }\n";
ptr += "    };\n";
ptr += "    xhr.send();\n";
ptr += "}\n";

```

```

ptr += "function startWashing() {\n";
ptr += " var xhr = new XMLHttpRequest();\n";
ptr += " xhr.open('POST', '/start_washing', true);\n";
ptr += " xhr.onreadystatechange = function() {\n";
ptr += "   if (this.readyState == 4 && this.status == 200) {\n";
ptr += "     document.getElementById('status').innerText = this.responseText;\n";
ptr += "   }\n";
ptr += " };\n";
ptr += " xhr.send();\n";
ptr += "}\n";
ptr += "function updateLiquidInfo() {\n";
ptr += " var liquid = document.getElementById('liquidSelect').value;\n";
ptr += " var info = ";\n";
ptr += " if (liquid === 'Liquid1') { info = 'Information about liquid: VIGON
US\n' +\n";
ptr += "           'Density at 20°C - 0.99 g/cm3\n' +\n";
ptr += "           'Boiling point - 165 - 212°C\n' +\n";
ptr += "           'pH - 11.3\n' +\n";
ptr += "           'Purification temperature - 40 - 60°C\n' +\n";
ptr += "           'Solubility in water - soluble\n' +\n";
ptr += "           'Solution concentration - 15 - 30%\n';\n";
ptr += "           }\n";
ptr += " if (liquid === 'Liquid2') { info = 'Information about liquid: ZESTRON
FA+\n' +\n";
ptr += "           'Density at 20°C - 0.99 g/cm3\n' +\n";
ptr += "           'Boiling point - 162 - 190°C\n' +\n";
ptr += "           'pH - 10.4\n' +\n";
ptr += "           'Purification temperature - 40 - 55°C\n' +\n";
ptr += "           'Solubility in water - soluble\n' +\n";

```

```

ptr += "                'Solution concentration - ready solution';\n";
ptr += "                }\n";
ptr += "    document.getElementById('liquidInfo').innerText = info;\n";
ptr += "    }\n";
ptr += "setInterval(fetchData, 2000);\n";
ptr += "setInterval(updateVideo, 500);\n";
ptr += "</script>\n";
ptr += "</head>\n";
ptr += "<body onload=\"fetchData(); updateVideo();\">\n";
ptr += "<h1>Monitoring Control System</h1>\n";
ptr += "<h3>WiFi mode (localIP)</h3>\n";
ptr += "<div id='sensorContainer'>\n";
ptr += "    <p>Temperature: <span id='temp'></span></p> <p>Temperature state:
<span id='tempState'></span></p>\n";
ptr += "    <p>Water level: <span id='waterLevel'></span></p> <p>Water level
state: <span id='waterState'></span></p>\n";
ptr += "    <p>Valve state: <span id='valveState'></span></p>\n";
ptr += "    <select id='liquidSelect' onchange='updateLiquidInfo()'>\n";
ptr += "        <option value='Liquid1' class='liquidOption'>VIGON US</option>\n";
ptr += "        <option value='Liquid2' class='liquidOption'>ZESTRON
FA+</option>\n";
ptr += "    </select>\n";
ptr += "    <p id='liquidInfo'></p>\n";
ptr += "    <button id='openValveButton' onclick='openValve()'>Открыть
ВЕНТИЛЬ</button>\n";
ptr += "    <button id='closeValveButton' onclick='closeValve()'>Закреть
ВЕНТИЛЬ</button>\n";
ptr += "    <button onclick='startWashing()'>Запуск отмывки</button>\n";
ptr += "    <p id='status'></p>\n";

```

```

ptr += "</div>\n";
ptr += "<div id=\"video-container\">\n";
ptr += "<img id='video' src='/camera' width='320' height='240'>\n";
ptr += "</div>\n";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}

```

ЛІСТИНГ А.3 – Функція handleRoot()

```

void handleRoot() {
    server.send(200, "text/html", SendHTML());
}

```

ЛІСТИНГ А.4 – Функція handleSensorData()

```

void handleSensorData(AsyncWebServerRequest *request) {
    tempSensor();
    soundSensor();
    String json = "{";
    json += "\"temperature\":" + String(temp) + ",";
    json += "\"tempState\":" + String(tempState) + ",";
    json += "\"distance\":" + String(distanceCm) + ",";
    json += "\"waterState\":" + String(waterState) + ",";
    json += "\"valveState\":" + String(valveState);
    json += "}";
    request->send(200, "application/json", json);
    saveToDB();
}

```

ЛІСТИНГ А.5 – Функція saveToDB()

```

void saveToDB()
{
    String response = "temperature=" + String(temp) + "&distance=" +
String(distanceCm) + "&dht_state=" + String(tempState) + "&ultrasonic_state=" +
String(waterState);
    HTTPClient http;
    http.begin(URL);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpResponseCode = http.POST(response);
    if (httpResponseCode > 0) {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
    } else {
        Serial.print("Error code: ");
        Serial.println(httpResponseCode);
    }
    http.end();
}

```

ЛІСТИНГ А.6 – Функція handleCameraStream()

```

void handleCameraStream(AsyncWebServerRequest *request){
    camera->oneFrame();
    AsyncWebServerResponse *response = request-
>beginChunkedResponse("image/bmp", [](uint8_t *buffer, size_t maxLen, size_t index)
-> size_t {
        static size_t headerSize = BMP::headerSize;
        static size_t dataIndex = 0;
        if (index < headerSize) {

```

```

    size_t len = min(maxLen, headerSize - index);
    memcpy(buffer, bmpHeader + index, len);
    return len;
}
size_t remainingData = camera->xres * camera->yres * 2 - dataIndex;
if (remainingData > 0) {
    size_t len = min(maxLen, remainingData);
    memcpy(buffer, camera->frame + dataIndex, len);
    dataIndex += len;
    return len;
} else {
    dataIndex = 0;
    return 0;
}
});
request->send(response);
}

```

Лістинг А.7 – Функції handleOpenValve() та handleCloseValve()

```

void handleOpenValve(AsyncWebServerRequest *request) {
    if (distanceCm <= 5) {
        request->send(200, "text/plain", "Уровень воды слишком высокий для
открытия вентиля!");
        return;
    }
    valve.write(180); // Открыть вентиль
    valveState = 1;
    request->send(200, "text/plain", "Вентиль открыт...");
}

```

```
void handleCloseValve(AsyncWebServerRequest *request) {
    valve.write(0); // ЗАКРЫТЬ ВЕНТИЛЬ
    valveState = 0;
    request->send(200, "text/plain", "ВЕНТИЛЬ ЗАКРЫТ...");
}
```

ЛІСТИНГ А.8 – Функція handleStartWashing()

```
void handleStartWashing(AsyncWebServerRequest *request) {
    if (temp < 15 || distanceCm > 5) {
        request->send(200, "text/plain", "Невозможно начать подготовку:
температура должна быть больше 15 градусов, а уровень воды меньше 5 см и
больше 1 см.");
        return;
    }
    if (washingInProgress) {
        request->send(200, "text/plain", "Уже идет подготовка ванны для
ОТМЫВКИ!");
        return;
    }
    washingInProgress = true;
    startTime = millis();
    stepperRotate();
    stepperTicker.once(10, stepperStop); // Начать процесс
    request->send(200, "text/plain", "Запущена подготовка ванны для ОТМЫВКИ!");
}
```

ДОДАТОК Б
Демонстраційний матеріал

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра КІТАР

КВАЛІФІКАЦІЙНА РОБОТА

На тему: Розроблення системи автоматизації для моніторингу роботи
приладобудівного приміщення з використанням ESP32

Виконав:
ст. гр. АКТАКІТ-20-1
Юрченко О.Д.

Керівник:
доц. каф. КІТАР
Бронніков А.І.

Актуальність роботи обумовлена високим попитом на системи моніторингу для забезпечення успішної діяльності підприємств. Приладобудівне виробництво є основою для багатьох галузей, виготовляючи необхідні електронні прилади та обладнання. Впровадження моніторингових систем підвищує якість продукції, скорочує час виробничих операцій та забезпечує надійну роботу виробництва.

Мета роботи – підвищення ефективності роботи виробничого процесу за рахунок розробки макету системи моніторингу виробничого процесу з використанням ESP32.

Об'єкт розробки – процес моніторингу параметрів рівня води та температури для відмивання друкованих плат, а також присутності людини на пропускному пункті приладобудівного приміщення.

Предмет розробки – система автоматизації моніторингу роботи приладобудівного приміщення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих систем моніторингу на виробництві;
- провести вибір основних елементів та технічних засобів для розробки системи моніторингу;
- розробити схему підключення системи;
- зібрати фізичний макет системи;
- розробити програмне забезпечення для системи.

Система моніторингу на виробництві

Особливості систем моніторингу на виробництві:

- надійність виробничих процесів;
- робота в реальному часі;
- масштабованість;
- інтеграція з системами управління;
- аналітика та звітність;
- сповіщення та реагування;
- простота використання.

Задачі системи моніторингу процесу ультразвукової відмивки ДП

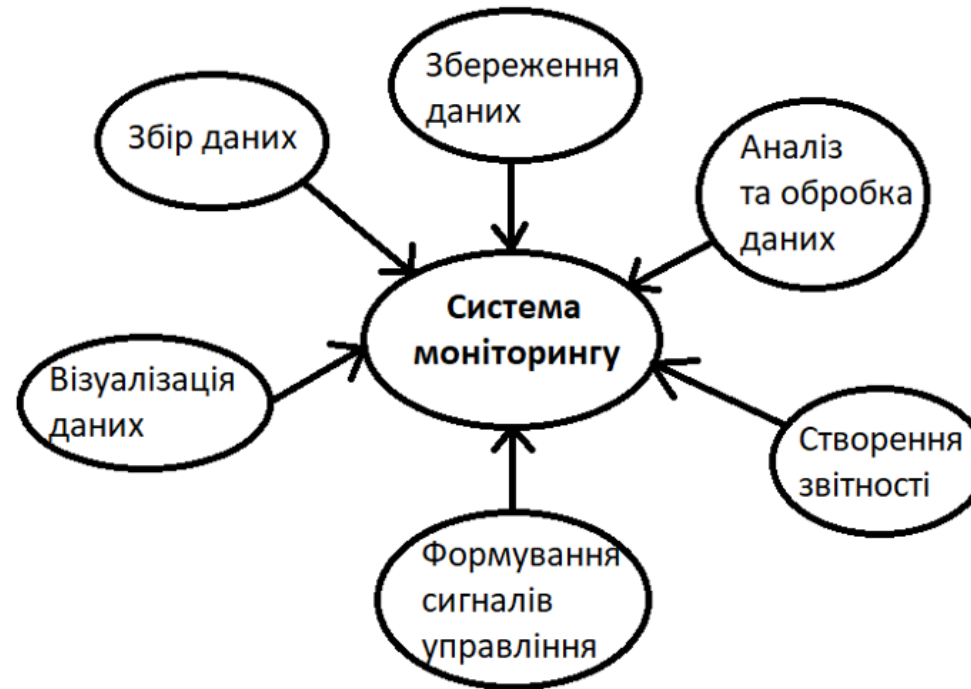


Рисунок 1 – Задачі, які повинна виконувати система моніторингу

Структурна схема системи моніторингу

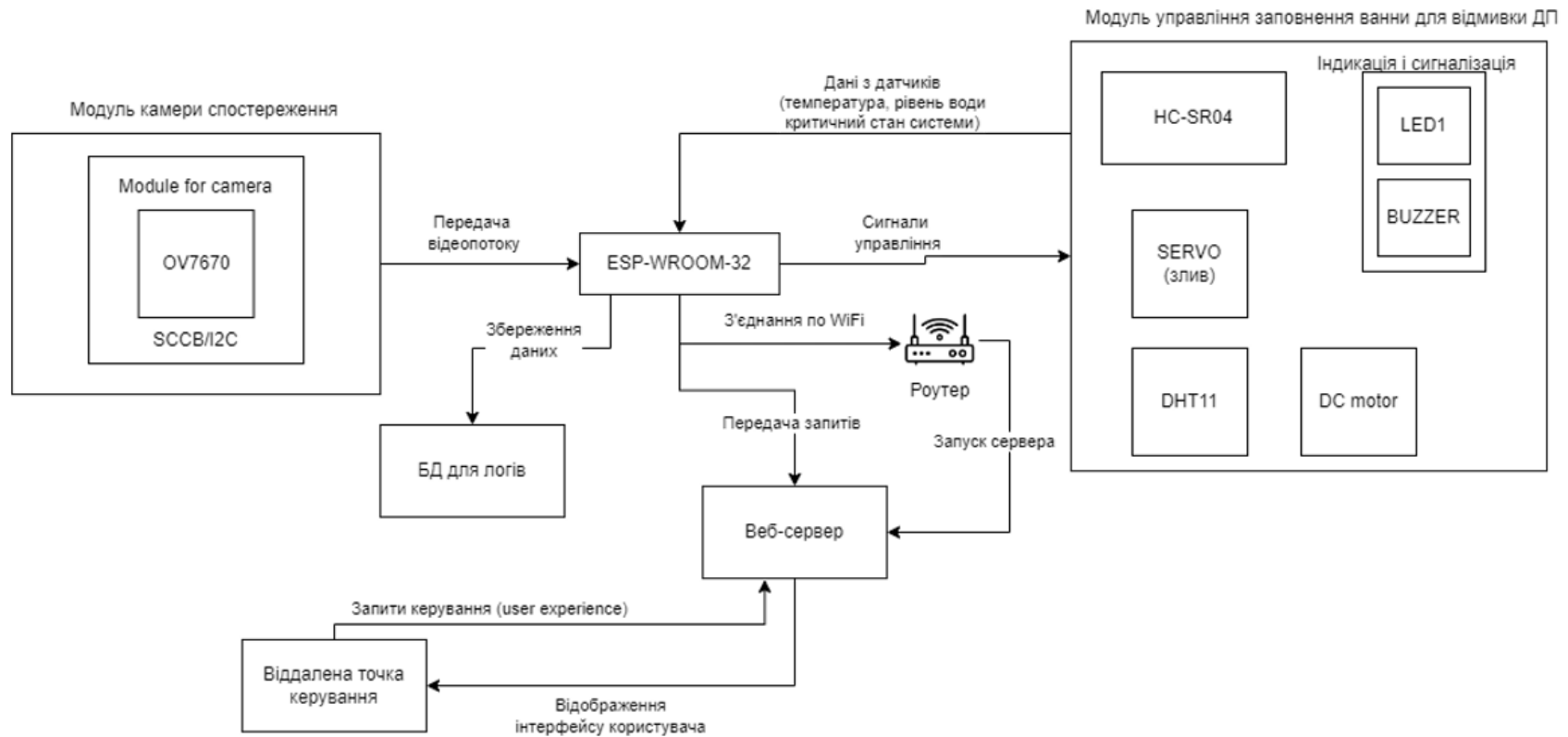


Рисунок 2 – Структурна схема розроблюваної системи

Апаратна складова системи



Рисунок 3 – ESP-WROOM-32

Основний модуль/орган керування системою, який підходить для розробки IoT продуктів.



Рисунок 4 – Модуль камери OV7670

Камера спостереження, яка забезпечує функціональність однокристальної VGA-камери та процесора обробки зображень.

Апаратна складова системи



Рисунок 5 – Датчик температури та вологості DHT11

Датчик, який вимірює температуру м'ячочої рідини в ванні.



Рисунок 6 – Ультразвуковий датчик відстані HC-SR04

Датчик вимірювання відстані до найближчої перешкоди. Використовується як заміна датчику рівня води.



Рисунок 7 – Серводвигун

Серводвигун використовується у якості вентиля зливу м'ячочої рідини у ванну.

Апаратна складова системи



Рисунок 8 – Електромотор

Електродвигун використовується для змішування розчину з водою для ультразвукової відмивки друкованих плат

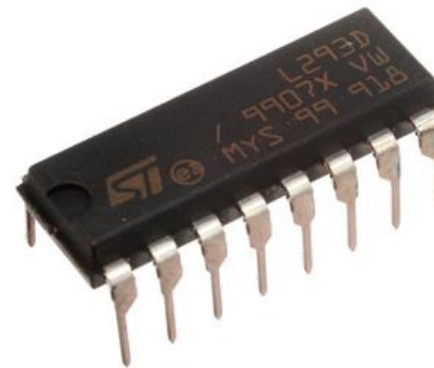


Рисунок 9 – Драйвер двигунів L293D

Драйвер двигунів використовується для підключення електромотора до плати ESP-WROOM-32

Апаратна складова системи



Рисунок 10 – Світлодіод



Рисунок 11 – Активний
п'єзодинамік

Світлодіод та п'єзодинамік використовуються в системі в якості елементів модуля індикація та сигналізації. Світлодіод відповідає за візуальне відображення критичних станів показників температури і рівня води, а п'єходинамік – звукова сигналізація.

Підключення апаратних елементів системи

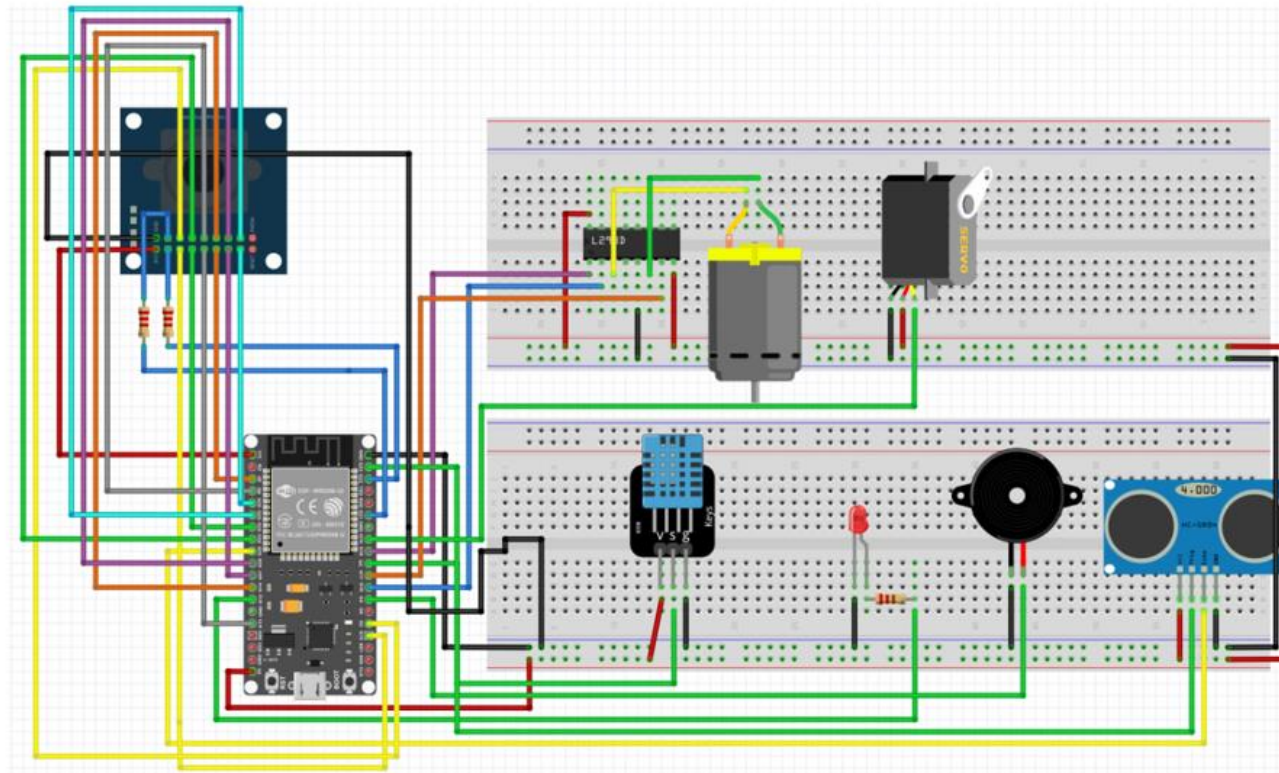


Рисунок 12 – Схема підключення

Збирання фізичного макету

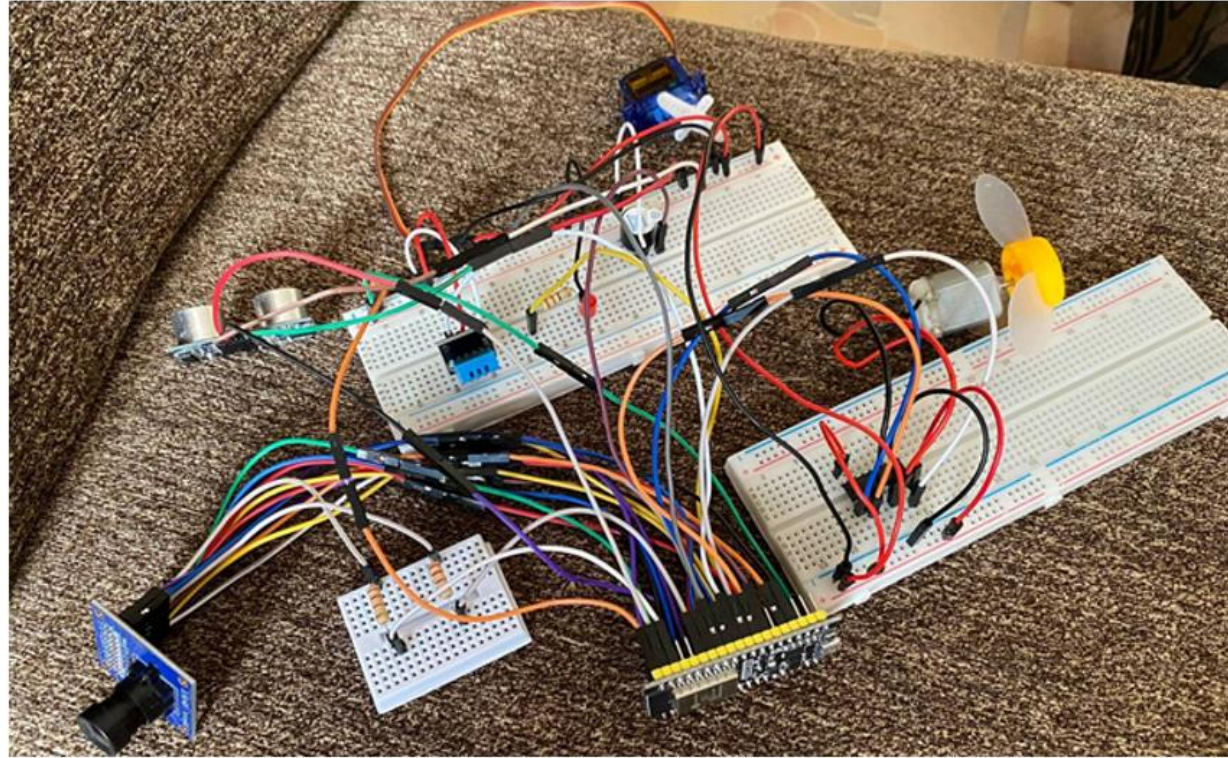


Рисунок 13 – Фізичний макет системи

Блок-схема роботи системи

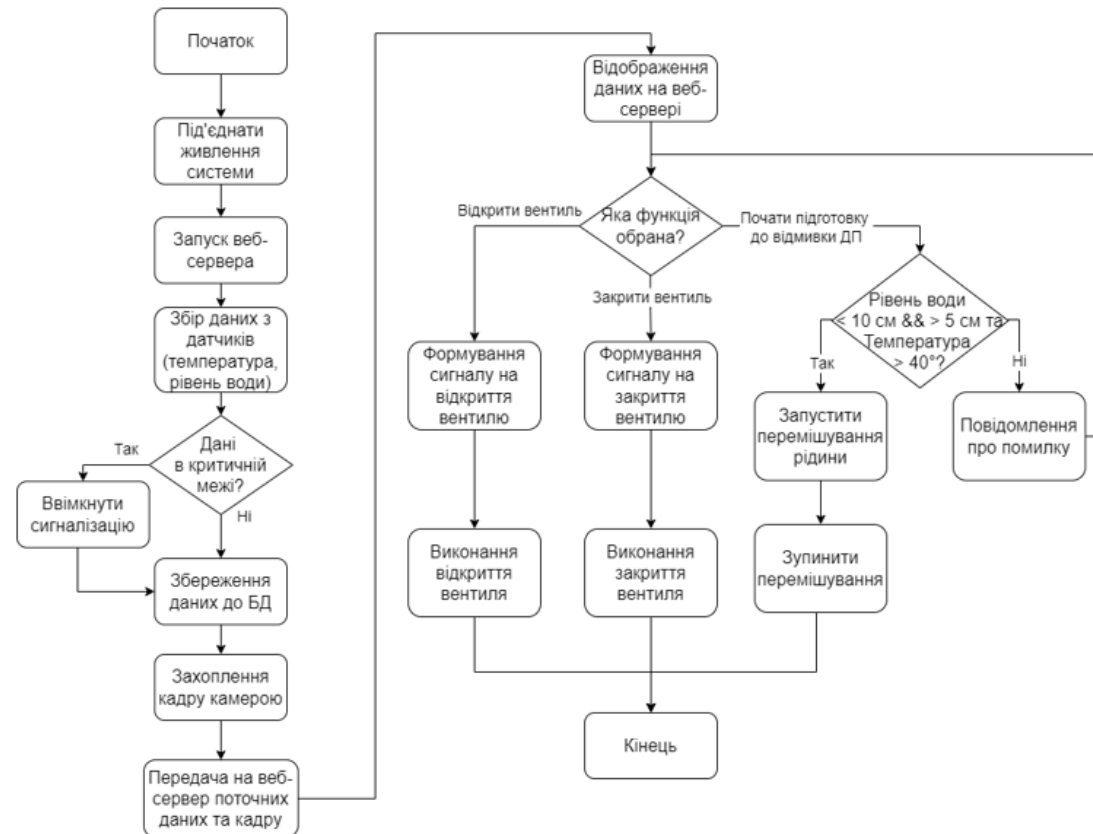


Рисунок 14 – Алгоритм роботи системи

Вибір технологій для розробки ПЗ

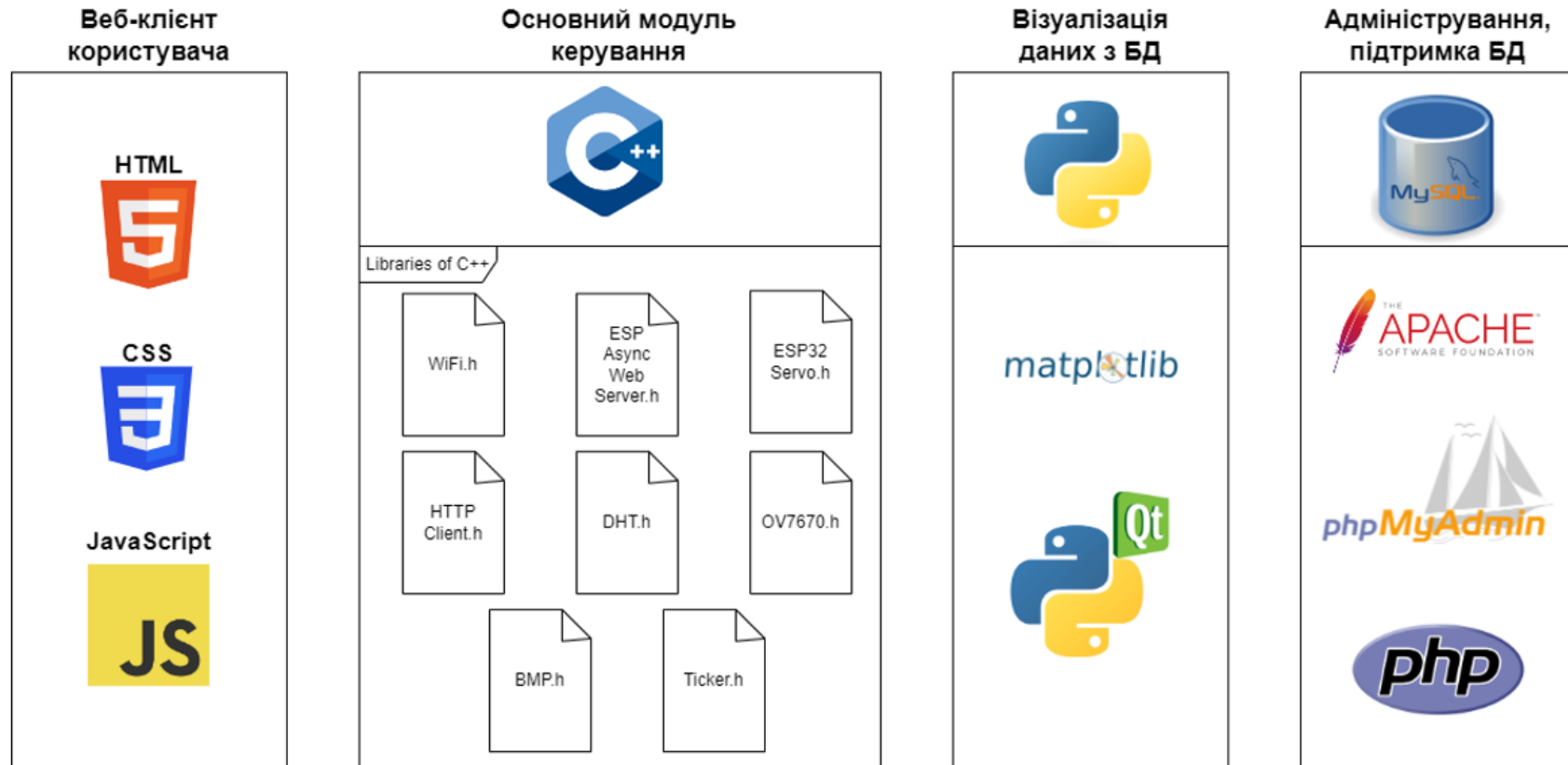


Рисунок 15 – Технології та технічні засоби

Результати роботи

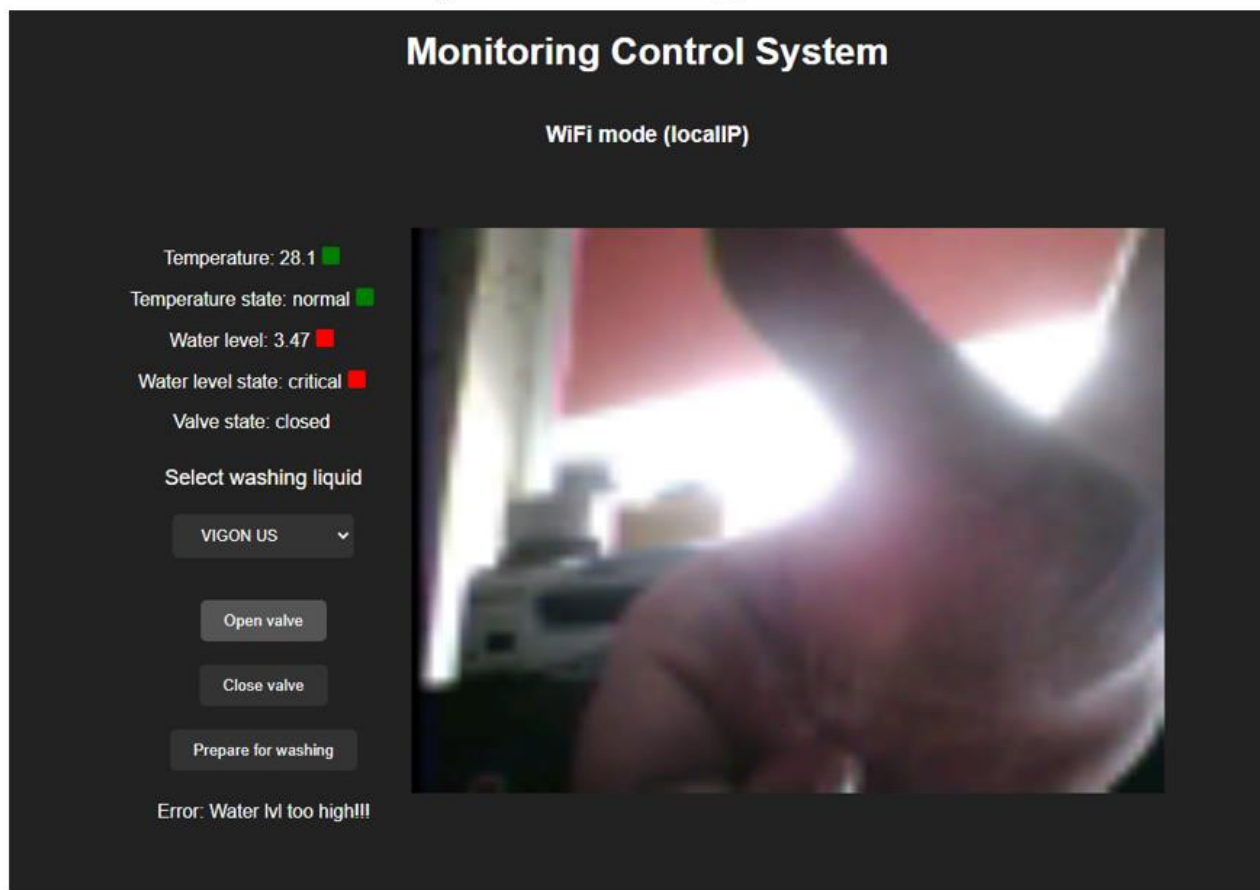


Рисунок 16 – Інтерфейс веб-сервера

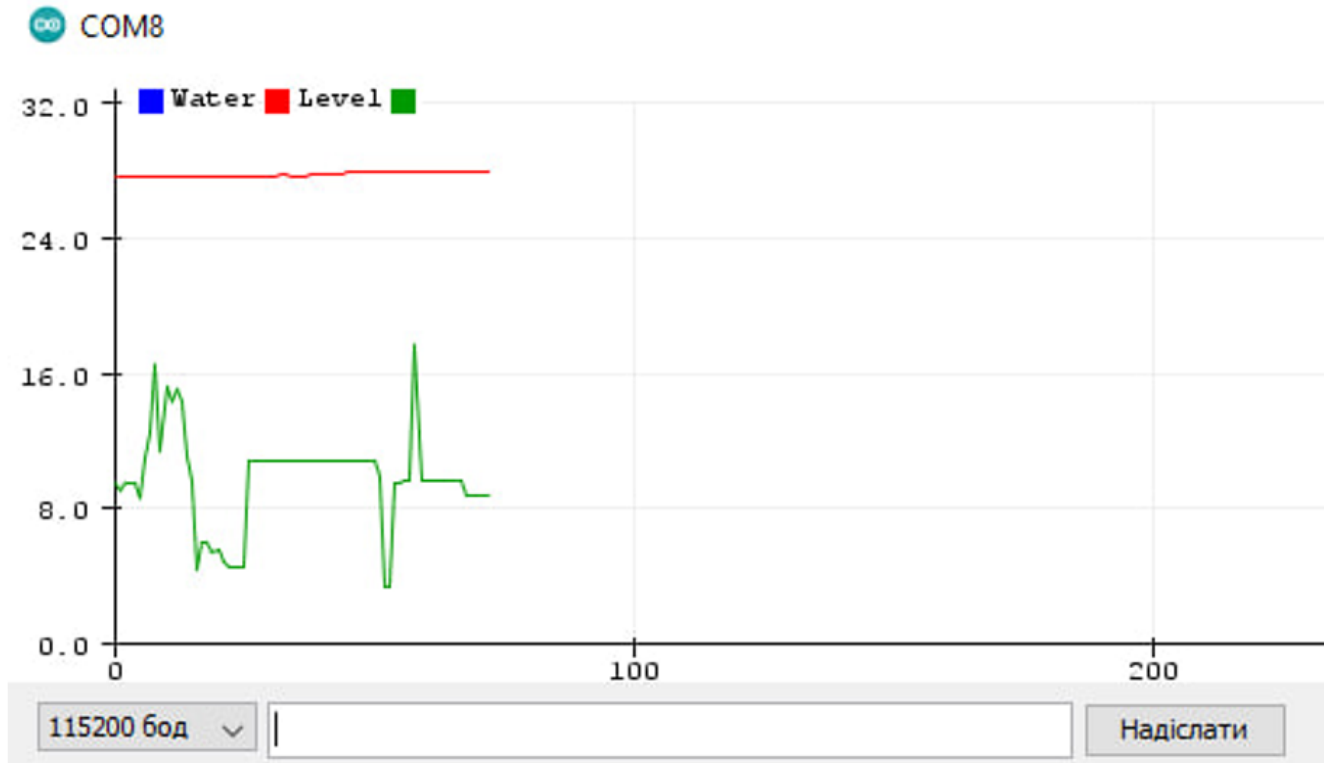


Рисунок 17 – Фіксація даних з датчиків

The screenshot displays the phpMyAdmin interface for a MySQL database. The left sidebar shows the database structure, including 'sensors_db' and its table 'sensors_log'. The main area shows the table view for 'sensors_log' with the following data:

	id	temperature	distance	dht_state	ultrasonic_state	log_time
<input type="checkbox"/>	1	19	6	0	0	2024-05-05 13:56:03
<input type="checkbox"/>	419	24	3	0	1	2024-05-05 16:46:15
<input type="checkbox"/>	420	24	7	0	0	2024-05-05 16:46:20
<input type="checkbox"/>	421	24	5	0	0	2024-05-05 16:46:25
<input type="checkbox"/>	422	24	12	0	0	2024-05-05 16:46:30
<input type="checkbox"/>	426	22	79	0	0	2024-05-10 22:10:47
<input type="checkbox"/>	427	22	5	0	0	2024-05-10 22:10:52
<input type="checkbox"/>	429	22	18	0	0	2024-05-10 22:11:02
<input type="checkbox"/>	430	22	19	0	0	2024-05-10 22:11:07
<input type="checkbox"/>	431	23	16	0	0	2024-05-10 22:11:12
<input type="checkbox"/>	432	22	17	0	0	2024-05-10 22:11:17
<input type="checkbox"/>	433	22	6	0	0	2024-05-10 22:11:22

Рисунок 18 – База данных MySQL

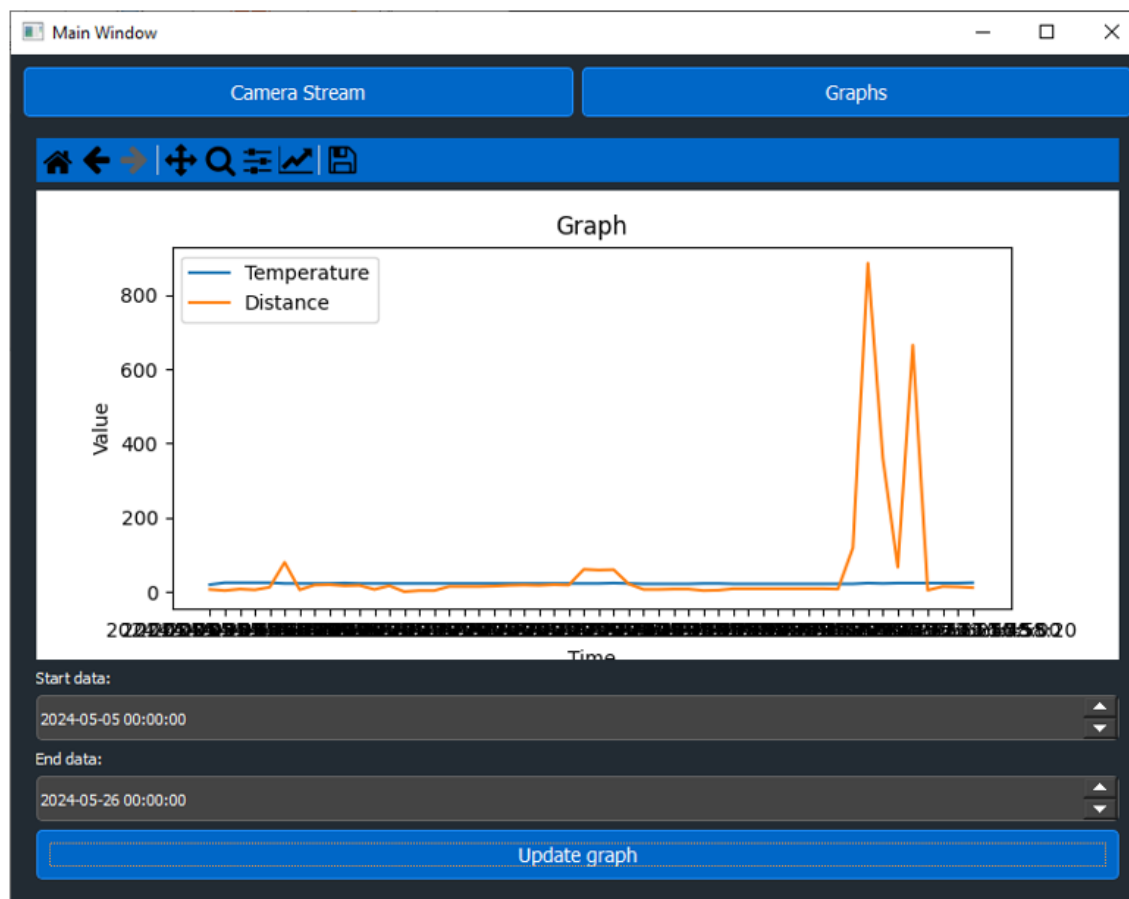


Рисунок 19 – Візуалізація даних з БД

Висновки

У ході кваліфікаційної роботи було проведено аналіз сучасного виробництва, розглянуті основні тенденції розвитку. Було проаналізовано впровадження нових систем та технологій, які забезпечують надійність роботи на підприємстві.

Були виконані наступні завдання, які ставилися:

- вивчено структуру сучасного виробництва;
- вивчено предметну область системи моніторингу виробничих процесів;
- проаналізовано та виконано порівняння існуючих рішень впровадження систем моніторингу;
- обрано апаратне забезпечення та технічні засоби для розробки системи моніторингу;
- розроблена схема підключення компонентів системи
- розроблена структурна схема;
- зібрано фізичний макет системи моніторингу;
- розроблено програмне забезпечення для керування системою моніторингу та відображення даних з БД у вигляді графіку.

У результаті виконання кваліфікаційної роботи було виконано основну мету, а саме розроблено макет та програмне забезпечення системи моніторингу виробничого процесу, а саме моніторингу параметрів рівня води, температури для ультразвукової відмивки друкованих плат.

Для подальшої розробки нових версій системи можна використати готові рішення на базі IoT та підвищити ефективність роботи системи завдяки впровадженню нових модулів.

