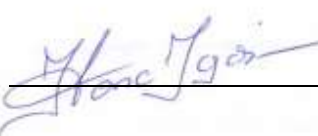
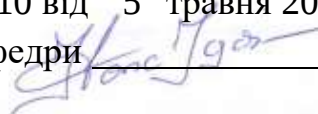


**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«БЕРЕЖАНСЬКИЙ ФАХОВИЙ КОЛЕДЖ НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ»
АГРОІНЖЕНЕРНЕ ВПДЛЕННЯ**

**Звіт
Студентського наукового гуртка
«Технології ремонту автомобільної техніки»
за 2024-2025 н.р.**

Викладач:  Igor
ФЛЬОНЦ

Розглянуто і схвалено на засіданні
Кафедри автомобільного транспорту
Протокол № 10 від "5" травня 2025 р.
Завідувач кафедри  Igor
ФЛЬОНЦ

м. Бережани 2025 р.

Мета і задачі досліджень.

Мета роботи – залучення студентів до наукової роботи, сприяння їх розвитку професійних навичок, пошуку нестандартних рішень у вирішенні практичних завдань.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- Формулювати задачу, технічні характеристики .
- Обґрунтування вибору запропонованої конструкторської розробки.
- Подати будову пристрою для визначення ефективності роботи клімат-контролю автомобіля.
- Описати принцип роботи пристрою для визначення ефективності роботи клімат-контролю автомобіля.

Актуальність теми полягає у важливості визначення основних показників якості повітря, такі як температуру, вологість та рівень вуглекислого газу CO_2 . Їх впливу на самопочуття водія.

Об'єкт дослідження – показники якості повітря, такі як температуру, вологість та рівень вуглекислого газу CO_2 .

Предмет дослідження – пристрій для визначення ефективності роботи клімат-контролю автомобіля.

Методи дослідження. Теоретичні дослідження проводились з використанням основних положень вищої математики, теоретичної механіки, теорії машин і механізмів, диференціального та інтегрального числення, а також за допомогою використання штучного інтелекту.

Наукова новизна отриманих результатів. Обґрунтовано конструктивні і технологічні параметри пристрою для визначення ефективності роботи клімат-контролю автомобіля.

Ключові слова: ЯКІСТЬ ПОВІТРЯ, КЛІМАТ-КОНТРОЛЬ, ШТУЧНИЙ ІНТЕЛЕКТ, ВОЛОГІСТЬ, ТЕМПЕРАТУРА, РІВЕНЬ CO_2 .

1. Обґрунтування вибору запропонованої конструкторської розробки.

Проаналізувавши конструкторські рішення, які зустрічаються у технічній літературі в Україні та світі, а також пристроями, що зустрічаються на ринку України, можна охарактеризувати вимоги до пошуку оптимальних параметрів пристрою для визначення ефективності роботи клімат-контролю автомобіля:

- Цей пристрій повинен вимірювати в салоні автомобіля такі основні показники якості повітря, такі як температуру, вологість та рівень вуглекислого газу CO_2 .
- Послідовність лабораторного дослідження у трьох режимах. 1. стан повітря без використання клімат-контролю, 50% потужності і 100% потужності його потужності.
- Зібрана інформація повинна передаватись у хмару через інтернет ресурси. Для прикладу у Google Таблиці.
- Після чого зібрана інформація аналізується.

Основним завданням у проектуванні , пошуку нової інформації, програмуванні і вивченні нового матеріалу є широке залучення штучного інтелекту (ШІ). Для цього використовували: чат GPT 4, DeepSeek, Qwen 2.5, Grok 3.

2. Будова модуля індукційного нагріву системи підігріву пального дизельних двигунів.

Список компонентів:

2. ESP8266 (наприклад, NodeMCU).
3. DHT11 (для вимірювання температури та вологості).
4. MQ-135 (для вимірювання концентрації газу).
5. LCD 16x2 з I2C модулем (для відображення даних).
6. Резистори :
7. 10 кОм для DHT11 (опціонально, якщо датчик має вбудований резистор).
8. Проводи для підключення .
9. Блок живлення (3.3V або 5V).

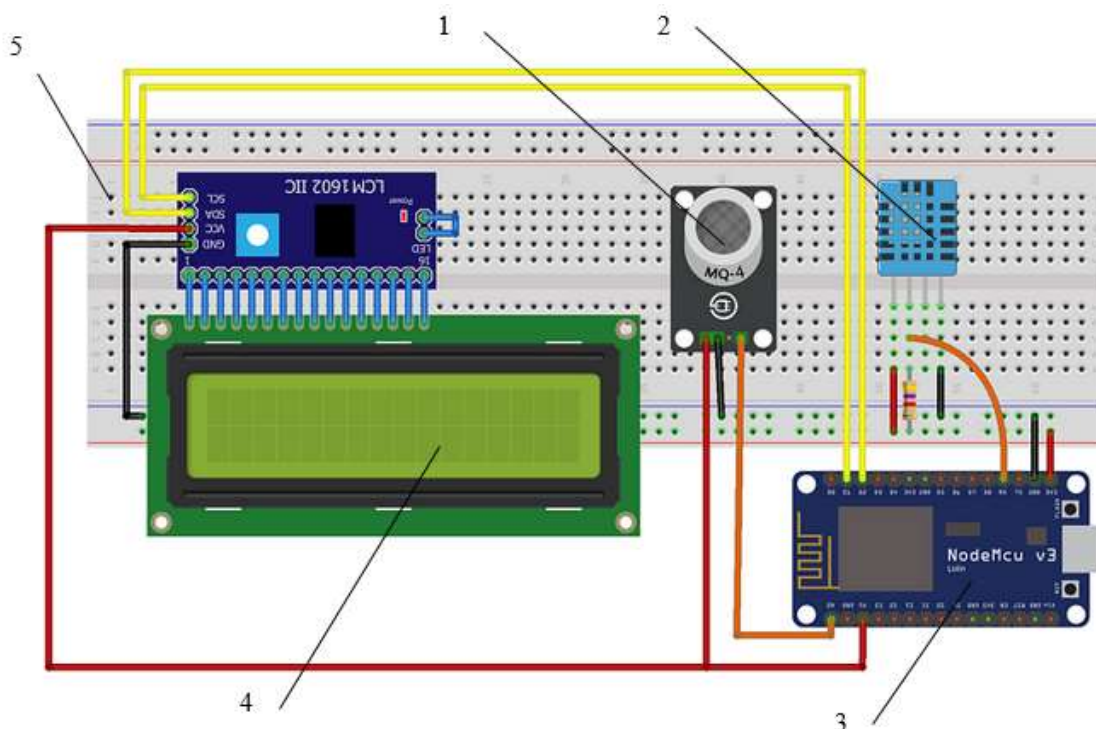


Рис. 1 Схема підключення елементів пристрою для визначення ефективності роботи клімат-контролю автомобіля:

1 – Модуль MQ-135 - датчик якості повітря; 2 – Модуль DHT11 датчик вологості та температури; 3 – і Модуль WiFi з ESP8266; 4. – Дисплей РК LCD 1602 16x2; 5 – Макетна плата.

Загальна схема підключення кожного з елементів.

1. ESP8266 (Вузловий мікроконтролер)

ПІНИ ESP8266	ПРИЗНАЧЕННЯ
3V3	Живлення +3.3V
ЗЕМЛЯ	Загальний провід (земля)
D3 (GPIO0)	Підключення до DHT11 (DATA)
A0	Підключення до MQ-135 (ANALOG)
D1 (GPIO5)	Підключення до SDA (LCD I2C)
D2 (GPIO4)	Підключення до SCL (LCD I2C)

2. DHT11 (Температура та Вологість)

ПІНІ DHT11	ПРИЗНАЧЕННЯ
Віртуальний кредитний центр (VCC)	Підключити до 3,3 В або 5 В
ЗЕМЛЯ	Підключити до GND
ДАНІ	Підключити до GPIO0 (D3)

Примітка: якщо використовуєте зовнішній резистор (10 кОм), підключіть його між VCC і DATA.

3. MQ-135 (Концентрація Газу)

ПІНІ MQ-135	ПРИЗНАЧЕННЯ
Віртуальний кредитний центр (VCC)	Підключити до 5V
ЗЕМЛЯ	Підключити до GND
AOUT	Підключити до A0 (аналоговий вхід)

4. LCD 16x2 з I2C модулем

ПІНІ LCD I2C	ПРИЗНАЧЕННЯ
Віртуальний кредитний центр (VCC)	Підключити до 5V
ЗЕМЛЯ	Підключити до GND
Адвентистська церковна ієрархія	Підключити до GPIO5 (D1)
СКЛ	Підключити до GPIO4 (D2)

3.4. Принцип роботи пристрою для визначення ефективності роботи клімат-контролю автомобіля.

1. Загальна ідея

Система збирає дані з датчиків (температура, вологість, концентрація газу), відображає їх на LCD-дисплеї та надсилає у Google Таблицю для подальшого аналізу або моніторингу. Усе це керується мікроконтролером ESP8266, який також забезпечує підключення до Wi-Fi.

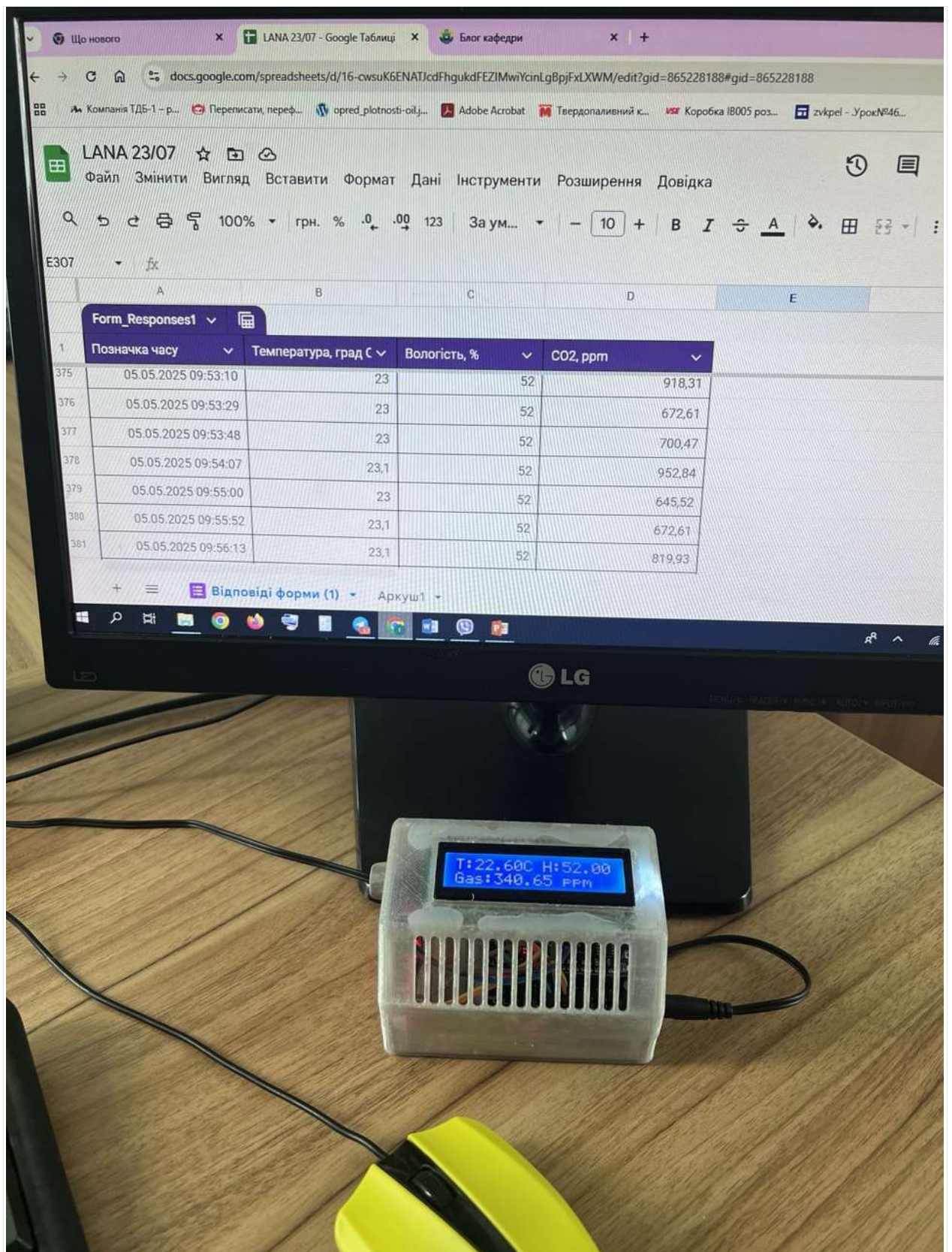


Рис. 2 Тестування пристрою для визначення ефективності роботи клімат-контролю автомобіля.

2. Принцип роботи системи

Крок 1: Ініціалізація

- ESP8266 запускається і виконує ініціалізацію:
- Підключається до Wi-Fi мережі за допомогою вказаних параметрів (`ssid` і `password`).
- Ініціалізує датчики (DHT11 і MQ-135) та LCD-дисплей.
- Встановлює з'єднання з сервером через HTTP-протокол.

Крок 2: Зчитування даних з датчиків

- DHT11 :
- DHT11 вимірює температуру (у °C) та вологість (у %).
- Дані зчитуються через цифровий пін GPIO0 (D3).
- MQ-135 :
- MQ-135 вимірює концентрацію газу (у ppm).
- Дані зчитуються через аналоговий пін A0.

Крок 3: Обробка даних

- Температура, вологість та концентрація газу обробляються програмою:
- Перевіряється коректність даних (наприклад, чи не повертає DHT11 значення `NaN`).
- Для MQ-135 може бути застосований коефіцієнт корекції (`K1`) для точнішого перетворення аналогового сигналу у ppm.

Крок 4: Відображення даних на LCD

- Дані відображаються на LCD-дисплеї:
- Перший рядок: температура (°C) та вологість (%).
- Другий рядок: концентрація газу (ppm).
- Приклад виведення:

T:23.5C H:45.6%

Gas:120.3 ppm

Крок 5: Надсилання даних у Google Таблицю

- Дані форматуються у JSON-формат:

```
json
{
  "temperature": 23.5,
  "humidity": 45.6,
  "gas": 120.3
}
```

- ESP8266 відправляє POST-запит на URL вашого Google Apps Script.

- Google Apps Script отримує дані, обробляє їх і записує у Google Таблицю.

Крок 6: Оновлення даних

- Процес повторюється кожні 15 секунд (задано через `delay(15000)`).

3. Роль кожного компонента ESP8266 (NodeMCU)

- Центральний контролер, який:
- Керує всіма підключеними пристроями (датчики, LCD).
- Забезпечує підключення до Wi-Fi.
- Відправляє дані на сервер через HTTP-запит.

DHT11

- Вимірює температуру та вологість.
- Працює за цифровим протоколом, тому дані легко зчитуються через бібліотеку `DHT`.

MQ-135

- Вимірює концентрацію газу (наприклад, CO₂, NH₃, алкоголь тощо).
- Працює через аналоговий вхід A0, який перетворюється у значення ppm за допомогою бібліотеки `MQ135`.

LCD 16x2 з I2C модулем

- Відображає поточні дані для користувача.
- I2C модуль зменшує кількість пінів, необхідних для підключення (лише SDA та SCL).

Google Apps Script

- Приймає дані у форматі JSON через POST-запит.
- Записує дані у Google Таблицю.
- Повертає відповідь про успішне виконання.

Google Таблиця

- Зберігає дані у структурованому вигляді:

Timestamp	Temperature	Humidity	Gas
-----	-----	-----	-----
2025-05-05 12:00:00	23.5	45.6	120.3

4. Послідовність роботи системи

1. Початок роботи :
 - ESP8266 запускається, ініціалізує всі компоненти та підключається до Wi-Fi.
2. Збір даних :

- Дані зчитуються з DHT11 (температура та вологість) та MQ-135 (концентрація газу).
3. **Обробка даних** :
 - Дані перевіряються на коректність.
 - Для MQ-135 застосовується коефіцієнт корекції.
 4. **Відображення даних** :
 - Дані виводяться на LCD-дисплей.
 5. **Надсилання даних** :
 - Дані форматуються у JSON та відправляються на сервер через HTTP POST-запит.
 6. **Запис у таблицю** :
 - Google Apps Script отримує дані, обробляє їх і записує у Google Таблицю.
 7. **Оновлення** :
 - Процес повторюється кожні 2 секунди.

5. Приклад даних у Google Таблиці

Після декількох циклів роботи таблиця може виглядати так:

TIMESTAMP	TEMPERATURE	HUMIDITY	GAS
2025-05-05 12:00:00	23.5	45.6	120.3
2025-05-05 12:00:15	23.6	45.7	121.0
2025-05-05 12:00:30	23.4	45.5	119.8

6. Скеч програми

```
- #include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <MQ135.h>

// LCD конфігурація
#define I2C_ADDR 0x27
#define ROWS 2
#define COLS 16

// Датчики
#define DHT_PIN 0 // Пін для DHT
#define MQ135_PIN A0 // Пін для MQ-135
float K1 = 25.0; // Коефіцієнт для MQ-135
```

```

LiquidCrystal_I2C lcd(I2C_ADDR, COLS, ROWS);
DHT dht(DHT_PIN, DHT11); // Ініціалізація DHT
MQ135 mq135(MQ135_PIN); // Ініціалізація MQ-135

// Wi-Fi налаштування
const char* ssid = "*****";
const char* password = "*****";

// URL Google Apps Script
const char* googleScriptURL =
"https://script.google.com/macros/s/AKfycby8lMD2tHS5CdqQnj-KYwFb7Roa_P_7_7yz1-
kqAm7zDWFZNJDluVZnq61zhyJeC1Ug/exec";

// Поля Google Apps Script
const char* entryTemperature = "entry.entryTemperature"; // Заміни на своє поле для
температури
const char* entryHumidity = "entry.entryHumidity"; // Заміни на своє поле для вологості
const char* entryGas = "entry.entryGas"; // Заміни на своє поле для газу

// Кнопки
#define BUTTON_BACKLIGHT_PIN D5 // GPIO14
#define BUTTON_DATA_PIN D6 // GPIO12

// Змінні для кнопок
bool lcdBacklightState = true; // Стан підсвічування
bool dataTransmissionEnabled = true; // Стан передачі даних
unsigned long buttonPressStartTime = 0; // Час натискання кнопки
bool buttonPressed = false; // Стан натискання кнопки
bool buttonHeld = false; // Стан утримання кнопки

void setup() {
  Serial.begin(115200);
  lcd.init();
  lcd.backlight(); // Увімкнути підсвічування за замовчуванням
  lcd.clear();
  dht.begin();

  // Налаштування кнопок
  pinMode(BUTTON_BACKLIGHT_PIN, INPUT_PULLUP); // Внутрішній підтягуючий
резистор
  pinMode(BUTTON_DATA_PIN, INPUT_PULLUP); // Внутрішній підтягуючий
резистор

  Serial.print("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected!");
  Serial.println(WiFi.localIP());
}

```

```

void loop() {
    // Перевірка кнопок
    checkButtons();

    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

    if (isnan(humidity) || isnan(temperature)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    int analogValue = analogRead(MQ135_PIN);
    float ppm = mq135.getPPM() * K1;

    // Виведення на LCD
    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print(temperature);
    lcd.print("C H:");
    lcd.print(humidity);
    lcd.print("%");

    lcd.setCursor(0, 1);
    lcd.print("Gas:");
    lcd.print(ppm);
    lcd.print(" ppm ");

    // Надсилення даних на Google Apps Script, якщо передача увімкнена
    if (dataTransmissionEnabled) {
        sendDataToServer(temperature, humidity, ppm);
    }

    delay(5000); // Затримка перед наступним зчитуванням
}

void checkButtons() {
    // Читання стану кнопки підсвічування
    if (digitalRead(BUTTON_BACKLIGHT_PIN) == LOW) {
        if (!buttonPressed) { // Кнопка натиснута
            buttonPressStartTime = millis();
            buttonPressed = true; // Зберегти стан натискання
        } else {
            // Перевірка часу натискання
            if (millis() - buttonPressStartTime >= 2000) { // Якщо натиснуто 2 секунди
                if (!buttonHeld) {
                    lcdBacklightState = !lcdBacklightState; // Змінити стан підсвічування
                    if (lcdBacklightState) {
                        lcd.backlight(); // Увімкнути підсвічування
                    } else {
                        lcd.noBacklight(); // Вимкнути підсвічування
                    }
                }
            }
        }
    }
}

```

```

    }
    Serial.println(lcdBacklightState ? "LCD backlight ON" : "LCD backlight OFF");
    buttonHeld = true; // Зберегти, що кнопка була утримана
  }
}
} else {
  buttonPressed = false; // Скинути стан натискання, якщо кнопка відпущена
  buttonHeld = false; // Скинути стан утримання кнопки
}

// Читання стану кнопки передачі даних
if (digitalRead(BUTTON_DATA_PIN) == LOW) {
  if (!buttonPressed) { // Кнопка натиснута
    buttonPressStartTime = millis();
    buttonPressed = true; // Зберегти стан натискання
  } else {
    // Перевірка часу натискання
    if (millis() - buttonPressStartTime >= 2000) { // Якщо натиснуто 2 секунди
      if (!buttonHeld) {
        dataTransmissionEnabled = !dataTransmissionEnabled; // Змінити стан передачі даних
        Serial.println(dataTransmissionEnabled ? "Data transmission ENABLED" : "Data
transmission DISABLED");
        buttonHeld = true; // Зберегти, що кнопка була утримана
      }
    }
  }
} else {
  buttonPressed = false; // Скинути стан натискання, якщо кнопка відпущена
  buttonHeld = false; // Скинути стан утримання кнопки
}
}

void sendDataToServer(float temperature, float humidity, float gas) {
  Serial.print("Sending data to Google Apps Script... ");

  WiFiClientSecure client;
  client.setInsecure(); // Ігнорувати SSL сертифікат

  HTTPClient http;
  if (http.begin(client, googleScriptURL)) {
    http.addHeader("Content-Type", "application/json");

    // Формування даних для відправки у форматі JSON
    String jsonData = "{\"temperature\":\"" + String(temperature) +
      "\",\"humidity\":\"" + String(humidity) +
      "\",\"gas\":\"" + String(gas) + "\"}";

    int httpResponseCode = http.POST(jsonData);

    if (httpResponseCode > 0) {
      Serial.printf("Response code: %d\n", httpResponseCode);
    }
  }
}

```

```

String response = http.getString();
Serial.println("Response: " + response);
} else {
Serial.printf("Error sending data. Response code: %d\n", httpResponseCode);
}

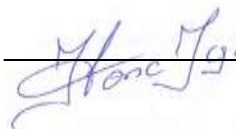
http.end();
} else {
Serial.println("Failed to connect to Google Apps Script.");
}
}

```

3. Висновки по розділу.

Проведені дослідження показали, що за допомогою ІІІ допомагає швидко шукати нову інформацію, проектувати, і реалізовувати на практиці технічні завдання. Пристрій для визначення ефективності роботи клімат-контролю автомобіля від початку до завершення проектувався за допомогою ІІІ. Корпус його накреслений у програмі SOLIDWORKS і був виготовлений за допомогою 3D друку.

Керівник гуртка

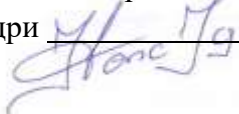
 Ігор ФЛЬОНЦ

Розглянуто і схвалено на засіданні

Кафедри автомобільного транспорту

Протокол № 10 від "5" травня 2025 р.

Завідувач кафедри

 Ігор ФЛЬОНЦ