

# ДОДАТОК А

## Апробація результатів наукових досліджень

---

Міністерство освіти і науки України



**NURE**

Харківський національний університет  
радіоелектроніки

### ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

**ADED-2023**

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-technologiy-avtomatizatsiyi-ta-mehatroniki-kitap>



<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2023

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки  
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(KITAP)



**ЗБІРНИК**  
студентських наукових статей  
«Автоматизація та приладобудування»  
«Automation and Development of Electronic Devices»  
**ADED-2023**  
(Випуск 2)  
[електронне видання]

Харків 2023

<b>Головний редактор</b>	<b>Невлюдов Ігор Шакирович</b> , доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.
<b>Редакційна колегія:</b>	<b>Филипенко Олександр Іванович</b> , доктор технічних наук, професор, декан факультету Автоматики та комп'ютеризованих технологій, Харківського національного університету радіоелектроніки. <b>Цимбал Олександр Михайлович</b> , доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки. <b>Андрусевич Анатолій Олександрович</b> , доктор технічних наук, професор, начальник Криворізького коледжу національного авіаційного університету <b>Косенко Віктор Васильович</b> , доктор технічних наук, професор, зам. директора Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості». <b>Замірець Микола Васильович</b> , доктор технічних наук, професор, директор Державного підприємства Науково-дослідного технологічного інституту приладобудування. <b>Свищ Володимир Митрофанович</b> , доктор технічних наук, професор, радник директора Державне науково-виробниче підприємство «Об'єднання Комунар». <b>Фомовська Олена Владиславівна</b> , кандидат технічних наук, доцент завідувач кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського. <b>Кухаренко Дмитро Володимирович</b> , кандидат технічних наук, доцент кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського <b>Демська Наталія Павлівна</b> , кандидат технічних наук, доцент кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки. <b>Фурманова Наталія Іванівна</b> , кандидат технічних наук, доцент, в.о. декана факультета Радіоелектроніки і телекомунікацій, Національного університету «Запорізька політехніка».
<b>Відповідальний редактор:</b>	<b>Євсєєв Владислав В'ячеславович</b> , доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.

процесів на підприємстві .....	
<i>А.В. Готовська</i>	
Підтримка прийняття рішень в технології проектування роботизованого виробничого процесу .....	213
<i>Я.В. Олінкевич</i>	
Впровадження egr-системи на виробництві .....	219
<i>М. Коваленко</i>	
Схема керування транспортними роботами на основі візуальних ознак .....	223
<i>В.К. Маковська</i>	
Контейнеризація та оркестрація: DOCKER та KUBERNETES .....	228
<i>Д.Р. Придятько</i>	
Огляд методів розпізнавання об'єктів за допомогою систем технічного зору .....	234
<i>А.А. Большаков</i>	
Розроблення архітектури SCADA-системи гнучкого виробництва та вибір апаратних засобів .....	239
<i>В.С. Головіна</i>	
Розроблення системи керування мобільним пошуково-рятувальним роботом .....	244
<i>Д.В. Мілько</i>	
Дослідження програмного методу визначення відстані до об'єкту за допомогою параметрів камери .....	250
<i>І.А. Манякін</i>	
Аналіз методів автоматичного розпізнавання осіб .....	254
<i>Ю.С. Візір</i>	
Автоматичне енергоефективне управління освітленістю з використанням кіберфізичних підходів в умовах виробництва .....	259
<i>В.І. Дульський</i>	
Методи оптимізації керуючих програм для верстатів з ЧПУ .....	264
<i>М.С. Карпов</i>	
Використання бездротових мереж для організації контролю в промисловості .....	269
<i>М.А. Пісклов</i>	
Алгоритми створення та оптимізації розкладу для загальноосвітніх навчальних закладів .....	275
<i>А.Ю. Губарь</i>	
Веб-додаток для моніторингу та управління запасами в 3D-друкарні .....	281
<i>І.А. Поддубняк</i>	
Аналіз сучасних візуальних SLAM систем в робототехніці .....	286
<i>Д.П. Редько</i>	
Технології транспортування вибухонебезпечних предметів за допомогою роботизованого пристрою .....	292
<i>В.О. Заїкін</i>	
Роботизовані системи та їх застосування у інноваційних методах виявлення та знешкодження вибухонебезпечних предметів .....	296
<i>К.О. Вадурін, А.С. Шандро</i>	
Розробка структури інформаційно-аналітичної система для збору, обробки та аналізу даних щодо використання енергетичних ресурсів багатоповерховою будівлею .....	302
<i>С.М. Гриценко</i>	
Аналіз систем контролю виготовлення 3D деталей на потоковому роботизованому виробництві .....	309

## РОЗРОБЛЕННЯ АРХІТЕКТУРИ SCADA-СИСТЕМИ ГНУЧКОГО ВИРОБНИЦТВА ТА ВИБІР АПАРАТНИХ ЗАСОБІВ

**А.А. Большаков**

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

Email: anton.bolshakov@nure.ua

**Анотація:** Дана стаття спрямована на розроблення архітектури SCADA-системи гнучкого виробництва. Для цього було проведено аналіз критеріїв, які потрібно враховувати при розробці архітектури SCADA-систем для гнучкого виробництва, що дозволило розробити архітектурну схему SCADA-системи. Обрані апаратні модулі для реалізації польового та PLC рівня, що дозволить розробити експериментальний макет для тестування програмного забезпечення SCADA-системи.

**Ключові слова:** SCADA, гнучке виробництво, архітектура SCADA-системи, модулі.

## DEVELOPMENT OF FLEXIBLE MANUFACTURING SCADA SYSTEM ARCHITECTURE AND SELECTION OF HARDWARE

**A. Bolshakov**

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av.,14

Email: anton.bolshakov@nure.ua

**Annotations:** This article is aimed at developing the SCADA system architecture of flexible production. For this, an analysis of the criteria that must be taken into account when developing the architecture of SCADA systems for flexible production was carried out, which made it possible to develop an architectural scheme of the SCADA system. Hardware modules are selected for the implementation of the field and PLC level, which will allow to develop an experimental layout for testing the software of the SCADA system.

**Key words:** SCADA, flexible production, SCADA system architecture, modules.

Розробка сучасних SCADA (Supervisory Control and Data Acquisition) систем в рамках концепції Industry 4.0 визначає нові стандарти для гнучкого виробництва, де технології збору та аналізу даних стають основою ефективності та інновацій. Ці системи стають ключовим елементом вдосконаленого моніторингу, контролю та оптимізації виробничих процесів, дозволяючи підприємствам швидко адаптуватися до змінних умов ринку та підтримувати гнучкість у виробництві. Інтеграція SCADA з технологіями Internet of Things (IoT), штучним інтелектом (AI) та аналітикою даних допомагає в реальному часі аналізувати та використовувати великі обсяги даних для прийняття стратегічних рішень. Забезпечення кібербезпеки та зменшення часу простою стають основоположними завданнями, що підкреслюють важливість розвитку та вдосконалення SCADA систем для сучасних виробничих середовищ в серії Industry 4.0 [1-5].

Дослідження та розробка архітектури SCADA-систем для гнучкого виробництва є актуальним завданням з таких причин:

- гнучкість та адаптивність, розробка адаптованої архітектури дозволяє SCADA-системам швидко та ефективно впроваджувати зміни у виробничих процесах, адаптуватися до нових вимог та забезпечувати гнучкість управління;

- інтеграція з іншими технологіями, архітектура повинна бути спроектована з урахуванням можливостей інтеграції з іншими технологіями Industry 4.0, такими як IoT, AI, хмарні рішення тощо, щоб забезпечити повністю організоване та зв'язане виробниче середовище;

- забезпечення ефективності, оптимізована архітектура дозволяє виробничим підприємствам максимально використовувати ресурси та забезпечує ефективність в управлінні та контролі виробничих процесів;

- керування великими обсягами даних, гнучке виробництво генерує значні обсяги даних. Архітектура SCADA повинна бути спроектована так, щоб ефективно обробляти, зберігати та аналізувати ці дані для прийняття важливих управлінських рішень;

- кібербезпека, розробка архітектури повинна враховувати сучасні стандарти та практики щодо кібербезпеки для захисту від можливих кібератак та збереження надійності виробничих систем.

- підтримка стандартів та протоколів, стандартизація та використання загальноприйнятих протоколів дозволяє підтримувати сумісність та інтеграцію з різними обладнаннями та системами виробництва.

- масштабованість та розширюваність, архітектура повинна бути розроблена з урахуванням майбутніх потреб та змін виробничих процесів, забезпечуючи можливість легкого розширення та модернізації системи з часом [6-8].

Дослідження та розробка архітектури SCADA-систем для гнучкого виробництва є ключовим етапом в досягненні сучасних стандартів управління та контролю в промисловості, відповідаючи викликам Industry 4.0 [9].

Розробка архітектури SCADA-систем для гнучкого виробництва вимагає врахування ряду критеріїв, щоб забезпечити оптимальну функціональність та відповідність сучасним вимогам. Основні критерії приведено в таблиці 1.

Таблиця 1 – Аналіз критеріїв, які потрібно враховувати при розробці архітектури SCADA-систем для гнучкого виробництва

Критерій	Опис
Гнучкість і адаптивність	Архітектура повинна бути здатна швидко адаптуватися до змін виробничих процесів та вимог ринку без значних витрат на модифікації
Інтеграція з іншими технологіями	Забезпечення можливості інтеграції з IoT, AI, хмарними технологіями та іншими системами, які підтримують концепції Industry 4.0
Кількість оброблених даних	Здатність обробляти великі обсяги даних у реальному часі та забезпечення їх ефективного аналізу для прийняття управлінських рішень
Безпека	Врахування сучасних стандартів кібербезпеки для захисту від потенційних загроз та забезпечення надійності системи
Швидкодія	Забезпечення високої швидкодії обміну даними для забезпечення реального часу при прийнятті управлінських рішень та контролю над процесами
Стандартизація і прозорість	Використання стандартів і прозорих протоколів для сприяння інтеграції з різними обладнаннями та підсистемами
Розширюваність	Можливість легкої розширення системи, яка дозволяє включати нові функції та апгрейди без значних зусиль
Легкість використання та інтуїтивний інтерфейс	Система повинна мати зрозумілий та легкий для використання інтерфейс для операторів та інших користувачів
Резервне копіювання та відновлення	Забезпечення можливості резервного копіювання та відновлення даних для уникнення втрати інформації при можливих збоях

Ці критерії взаємодіють між собою та їх правильний баланс дозволяє розробити SCADA-систему, яка відповідає вимогам гнучкого виробництва та концепціям Industry 4.0 [10]. Базуючись на вище перерахованих критеріях, в рамках даних досліджень було запропоновано наступну архітектурну схему SCADA-системи для гнучкого виробництва, яка представлена на рисунку 1.

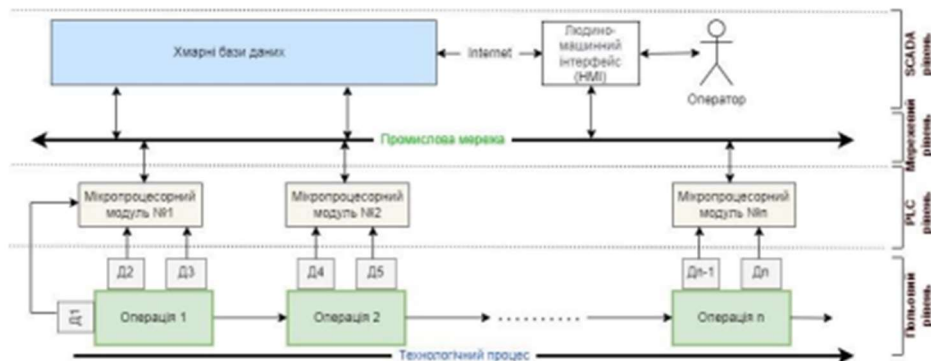


Рисунок 1 – Розроблена архітектурна схема SCADA-системи для гнучкого виробництва

Як можна бачити з рисунку 1, розроблена архітектурна схема SCADA-системи для гнучкого виробництва має чотири рівня:

польовий рівень (field level) – в контексті виробництва вказує на рівень системи автоматизації, де проводяться фізичні вимірювання та збір даних безпосередньо з об'єктів на виробничому майданчику. Цей рівень зазвичай включає в себе датчики, вимірювальні прилади, а також інші пристрої, які здійснюють збір реальних параметрів та величин;

PLC рівень (Programmable Logic Controller) – на виробництві вказує на використання спеціальних пристроїв або систем, які призначені для автоматизації та контролю над технологічними процесами в промисловості [11];

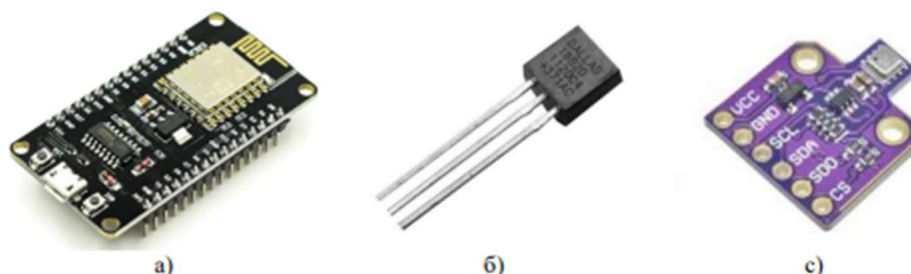
мережевий рівень – це рівень інфраструктури, де відбувається обмін даними між різними пристроями та системами виробництва. Цей рівень включає в себе мережеві технології та протоколи, які забезпечують зв'язок між пристроями польового та PLC рівнів і можуть також включати засоби комунікації з іншими вищими рівнями автоматизації [12];

SCADA рівень (Supervisory Control and Data Acquisition) – представляє собою вищий рівень системи автоматизації, який забезпечує збір, візуалізацію, контроль та управління даними та процесами виробництва. SCADA рівень включає в себе програмне забезпечення, яке дозволяє операторам та інженерам взаємодіяти з обладнанням, моніторити його стан, а також приймати рішення для ефективного управління виробничими процесами [13].

Розглянемо принцип роботи розробленої архітектурної схеми SCADA-системи для гнучкого виробництва. В ході виконання операцій над заготовкою, за допомогою масивів датчиків (Д1, Д2, ..., Дn) отримуємо дані з перебігу технологічного процесу в цій точці виробництва. Отримані дані надходять на мікропроцесорний модуль, в якому проводиться перетворення сигналу, аналіз і прийняття рішення або передача даних на рівень вище через промислову мережу. Дані через промислову мережу передаються в хмарне сховище даних, в якому відбувається їх аналіз, упорядкування за необхідними параметрами. Все це дозволяє спростити процес візуалізації виробничих даних для оператора, який може запросити інформацію про

протікання технологічного процесу в режимі реального часу, як через промислову мережу, так і через мережу глобального Інтернету [14-15].

Наступним кроком для реалізації, необхідно обрати апаратні модулі для створення макету, що дозволить провести тестування. Внаслідок чого, для реалізації польового та PLC рівня обрані наступні модулі та сенсори, які представлено на рисунку 2.



а) Модуль NODEMCU ESP8266 CH340 V3 LUA 3 WI-FI;  
 б) DS18B20 датчик температури;  
 в) Модуль датчика якості повітря BME680 I2C/5B

Рисунок 2 – Обрані апаратні модулі для реалізації польового та PLC рівня

Модуль NODEMCU ESP8266 CH340 V3 з Wi-Fi обраний для реалізації SCADA-системи в гнучкому виробництві з кількох ключових причин. По-перше, вбудований модуль Wi-Fi дозволяє забезпечити бездротовий доступ та збір даних з різних точок виробництва, спрощуючи інтеграцію в систему моніторингу. Зменшені розміри модуля роблять його ідеальним для вбудовування в різноманітне обладнання без значного впливу на його габарити. Економічна доступність ESP8266 робить його ефективним рішенням для гнучких виробничих систем з обмеженим бюджетом. Простота програмування за допомогою мови Lua та підтримка Arduino IDE роблять цей модуль доступним для розробників різного досвіду. З великою спільнотою розробників та активними ресурсами, NODEMCU ESP8266 забезпечує надійність та ефективність у створенні системи SCADA для гнучкого виробництва [16-17].

Висновки: В рамках проведеного дослідження було проведено аналіз критеріїв, які потрібно враховувати при розробці архітектури SCADA-систем для гнучкого виробництва. На базі проведеного аналізу було розроблено архітектурну схему SCADA-системи для гнучкого виробництва, яка складається з 4 рівнів. Це дозволило обрати апаратні модулі для реалізації польового та PLC рівня, що дозволить розробити експериментальний макет для тестування програмного забезпечення SCADA-системи.

#### ЛІТЕРАТУРА

1. Geeta Yadav, Kolin Paul. (2021). Architecture and security of SCADA systems: A review. International Journal of Critical Infrastructure Protection. Volume 34. <https://doi.org/10.1016/j.ijcip.2021.100433>.
2. Byres, E.: SCADA Security Basics: SCADA vs. ICS Terminology. Available at: <https://www.tofinosecurity.com/blog/scada-security-basics-scada-vs-ics-terminology>. Accessed: 21.11.2023.
3. Russel, J.: A Brief History of SCADA/EMS. Available at: <http://www.Scada history.com/>. Accessed: 21.11.2023

4. Tidrea, Alexandra, Adrian Korodi, and Ioan Silea. 2023. "Elliptic Curve Cryptography Considerations for Securing Automation and SCADA Systems" *Sensors* 23, no. 5: 2686. <https://doi.org/10.3390/s23052686>.
5. Rosa, L.; Freitas, M.; Mazo, S.; Monteiro, E.; Cruz, T.; Simoes, P. A Comprehensive Security Analysis of a SCADA Protocol: From OSINT to Mitigation. *IEEE Access* 2019, 7, 42156–42168.
6. Attar, H., & et al. (2022). Control System Development and Implementation of a CNC Laser Engraver for Environmental Use with Remote Imaging. *Computational Intelligence and Neuroscience*, 2022, Article ID 9140156, <https://doi.org/10.1155/2022/9140156>.
7. Khalid, M. S., Yevsieiev, V., Nevliudov, I. S., Lyashenko, V., & Wahid, R. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. *International Journal of Engineering Trends and Technology*, 70.1, 139-145.
8. Nevliudov, I., & et al. (2021). Development of a cyber design modeling declarative Language for cyber physical production systems, *J. Math. Comput. Sci.*, 11(1), 520-542.
9. Nevliudov, I., & et al. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive CyberDesign CPPS Development. *Advances in Dynamical Systems and Applications*, 16(2), 441-455.
10. Nevliudov, I., & et al. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. *International Journal of Emerging Trends in Engineering Research*, 8(10), 7465-7473.
11. Conveyor Belt Object Identification: Mathematical, Algorithmic, and Software Support / V. V. Yevsieiev, I. S. Nevliudov, S. S. Maksymova, M. A. O. Omarov, O. M. Klymenkoю // *Appl. Math. Inf. Sci.* 17, No. 6. - P. 1073-1088.
12. Невлюдов І.Ш. Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник / І.Ш. Невлюдов, А.О. Андрусевич, В.В. Євсєєв, С.С. Максимова, М.Г. Стародубцев, В.В.Невлюдова. Кривий Ріг: Криворізький коледж НАУ, 2018. 320 с.
13. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : монографія / І. Ш. Невлюдов, В. В. Євсєєв, А. О. Андрусевич, С. С. Максимова ; – Oktan Print – Prague. 2023. – 321 с.
14. Yevsieiev V., Bronnikov A. Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production. The 11th International scientific and practical conference «Development of scientific and practical approaches in the era of globalization» (USA, Boston, 28–30 September. 2020). P.221–226. DOI:10.46299/ISG.2020.II.II.
15. Nevliudov I., Omarov M., Yevsieiev V., Bronnikov A., Lyashenko V. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis // *International Journal of Emerging Trends in Engineering Research*. – 2020. – Vol. 8(10). – P. 7465-7473.
16. Yevsieiev, V. V., & Bronnikov, A. I. (2020). Development of databases interconnection "essences" information model for cyber-physical production systems additive cyber design creation automation. *Збірник Наукових Праць НУК*, №3. С.56-62. DOI [https://doi.org/10.15589/znp2020.3\(481\).7](https://doi.org/10.15589/znp2020.3(481).7).
17. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development. III International scientific-practical conference "Theory, science and practice" (Japan, Tokyo, 5–8 October 2020). P. 398–401. DOI: 10.46299/ISG.2020.II.III.

**Науковий керівник:** Цимбал Олександр Михайлович, д.т.н., професор кафедри КІТАР Харківського національного університету радіоелектроніки

## ДОДАТОК Б

### Основний файл прошивки

```
#include "secrets.h"
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include "WiFi.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_BMP280.h>

#define BMP_SCK (13)
#define BMP_MISO (12)
#define BMP_MOSI (11)
#define BMP_CS (10)x

Adafruit_BMP280 bmp; // I2C

// #define AWS_IOT_PUBLISH_TOPIC "esp32/pub"
#define AWS_IOT_PUBLISH_TOPIC "$aws/things/esp32/shadow/update"
#define AWS_IOT_SUBSCRIBE_TOPIC "esp32/sub"
// # define AWS_IOT_SUBSCRIBE_TOPIC
"$aws/things/esp32/shadow/update/delta"
```

```
#define FANpin 25
#define FANfrequency 2500
#define PWM_RES 8
const int MAX_DC = (int)(pow(2, PWM_RES) - 1);

#define DO_PIN 14 // ESP32's pin GPIO14 connected to DO pin of the MQ2
sensor

float temperature;
float pressure;
float altutude;
time_t seconds;
int brightness = 0;
int fadeAmount = 1;

const int detectin = 23; // or 4
const int soundinp = 32; // or 4

int detectState = 0;

int MICRO_PIN = 26; // KY-037 digital interface
int digitalVal; // digital readings
bool soundLimit;
bool gasPresence;
bool humanInSight;

WiFiClientSecure net = WiFiClientSecure();
PubSubClient client(net);
```

```
void connectAWS()
{
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.println("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  // Configure WiFiClientSecure to use the AWS IoT device credentials
  net.setCACert(AWS_CERT_CA);
  net.setCertificate(AWS_CERT_CRT);
  net.setPrivateKey(AWS_CERT_PRIVATE);
  // Connect to the MQTT broker on the AWS endpoint we defined earlier
  client.setServer(AWS_IOT_ENDPOINT, 8883);
  // Create a message handler
  client.setCallback(messageHandler);
  Serial.println("Connecting to AWS IOT");
  while (!client.connect(THINGNAME))
  {
    Serial.print(".");
    delay(100);
  }
  if (!client.connected())
  {
    Serial.println("AWS IoT Timeout!");
    return;
  }
}
```

```

// Subscribe to a topic
client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC);
Serial.println("AWS IoT Connected!");
setCurrentTime();
}

void publishMessage()
{
  StaticJsonDocument<200> doc;
  JsonObject state_reported = doc["state"].createNestedObject("reported");
  state_reported["pressure"] = pressure;
  state_reported["temperature"] = temperature;
  state_reported["approx_altitude"] = altutude;
  state_reported["sound"] = soundLimit;
  state_reported["gas"] = gasPresence;
  state_reported["human"] = humanInSight;
  state_reported["timestamp"] = seconds;
  char jsonBuffer[512];
  serializeJson(doc, jsonBuffer); // print to client
  client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
}

void messageHandler(char* topic, byte* payload, unsigned int length)
{
  Serial.print("incoming: ");
  Serial.println(topic);
  StaticJsonDocument<200> doc;
  deserializeJson(doc, payload);
  const char* message = doc["message"];
  Serial.println(message);
}

```

```

}

void setup()
{
  Serial.begin(115200);
  // Serial.begin(9600);
  pinMode(DO_PIN, INPUT);
  pinMode(MICRO_PIN, INPUT);
  while ( !Serial ) delay(100); // wait for native usb

  Serial.println(F("BMP280 test"));
  unsigned status;
  //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
  status = bmp.begin(0x76);
  if (!status) {
    Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
      "try a different address!"));
    Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(), 16);
    Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or
BMP 085\n");
    Serial.print("    ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print("    ID of 0x60 represents a BME 280.\n");
    Serial.print("    ID of 0x61 represents a BME 680.\n");
    while (1) delay(10);
  }

  /* Default settings from datasheet. */
  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode.
*/
    Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */

```

```

Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
Adafruit_BMP280::FILTER_X16, /* Filtering. */
Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */

connectAWS();
pinMode(detectin, INPUT);
pinMode(soundinp, OUTPUT);
ledcSetup(1, FANfrequency, PWM_RES);
ledcAttachPin(FANpin, 1);
}

void setCurrentTime() {
  configTime(5.5 * 3600, 0, "pool.ntp.org", "time.nist.gov"); //UTC

  Serial.print("Waiting for NTP time sync: ");
  time_t now = time(nullptr);
  while (now < 8 * 3600 * 2) {
    delay(500);
    Serial.print(".");
    now = time(nullptr);
  }
  Serial.println("");
  struct tm timeinfo;
  gmtime_r(&now, &timeinfo);
  Serial.print("Current time: "); Serial.print(asctime(&timeinfo));

}

```

```
void loop()
{

    if (bmp.readTemperature() < 34) {
        ledcWrite(1, 0);
    } else {
        ledcWrite(1, MAX_DC);
    }
    digitalVal = digitalRead(MICRO_PIN);

    if (digitalVal == HIGH)
    {
        soundLimit = true;
        // digitalWrite(soundinp, HIGH);
        digitalWrite(soundinp, HIGH);
        delay(100);
        digitalWrite(soundinp, LOW) ;
        delay(100);
        digitalWrite(soundinp, HIGH);
        delay(100);
        digitalWrite(soundinp, LOW) ;
        delay(100);
        digitalWrite(soundinp, HIGH);
        delay(100);
        digitalWrite(soundinp, LOW) ;
        Serial.printf("Sound is too loud");
    }
}
```

```
else
{
    soundLimit = false;
    digitalWrite(soundinp, LOW) ;
    digitalWrite(detectState, LOW) ;
    Serial.printf("Sound is fine");
}

Serial.println();
detectState = digitalRead(detectin);

if (detectState == HIGH) {
    digitalWrite(soundinp, HIGH);
//    digitalWrite(soundinp, LOW);
    humanInSight = true;
    Serial.printf("Someone detected in sight");
} else {
    digitalWrite(soundinp, LOW) ;
    detectState = 0;
    humanInSight = false;
    digitalWrite(detectState, LOW) ;
    Serial.printf("No one detected in sight");
}

Serial.println();
int gasState = digitalRead(DO_PIN);

if (gasState == HIGH) {
    //    digitalWrite(soundinp, HIGH); //потом вернуть
    digitalWrite(soundinp, LOW);
```

```
    ledcWrite(1, 0);
    gasPresence = false;
    Serial.println("The gas is NOT present");
}
else {
    gasPresence = true;
    Serial.println("The gas is present");
    digitalWrite(soundinp, HIGH);
    ledcWrite(1, MAX_DC);
}
seconds = time (NULL);
Serial.print ("Timestamp " );
Serial.print(seconds);
Serial.println();
Serial.print(F("Temperature = "));
Serial.print(bmp.readTemperature());
Serial.println(" *C");
temperature = bmp.readTemperature();
Serial.print(F("Pressure = "));
Serial.print(bmp.readPressure());
Serial.println(" Pa");
pressure = bmp.readPressure();
Serial.print(F("Approx altitude = "));
Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */
Serial.println(" m");
altutude = bmp.readAltitude(1013.25);
Serial.println();

delay(500);
```

```
publishMessage();
```

```
client.loop();
```

```
delay(1000);
```

```
if (WiFi.status() != WL_CONNECTED)
```

```
{
```

```
  Serial.println("Reconnecting");
```

```
  delay(10000);
```

```
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
```

```
}
```

```
}
```

**ДОДАТОК В****Демонстраційний матеріал у вигляді презентації**

