

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Предметно-циклова комісія інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНИЙ

Голова ПЦК \_\_\_\_\_

спеціаліст в/к Сабанов С.О.

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА

**СТВОРЕННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ  
ГОДИННИК-МЕТЕОСТАНЦІЯ НА БАЗІ МІКРОКОНТРОЛЛЕРУ ESP8266**

Виконав

ст. гр. КІ -119К9 \_\_\_\_\_

Бибик Т.І.

Керівник

ст. викладач \_\_\_\_\_

Дереза К.В.

Запоріжжя

2023

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Предметно-циклова комісія інформаційних технологій

ЗАТВЕРДЖУЮ

Голова ПЦК спеціаліст в/к  
Сабанов С.О. \_\_\_\_\_  
“” 2023 року

ЗАВДАННЯ

ВИПУСКНОЇ РОБОТИ МОЛОДШОГО СПЕЦІАЛІСТА

Студента гр. KI-119K9

Спеціальності: 123 – Комп'ютерна інженерія

Бибику Тарасу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема: «Створення програмно-апаратного комплексу годинник  
метеостанція на базі мікроконтролеру ESP8266»

затверджена наказом по інституту: № \_\_\_\_\_ від 2023 року

2. Термін здачі студентом закінченої роботи: 2023 року

3. Перелік питань, що підлягають розробці:

1. Зібрати літературу та документацію присвячену тематиці  
випускної роботи;

2. Провести бесіду з керівництвом підприємства з приводу  
випускної роботи та впровадження як розробку;

3. Розглянути та проаналізувати аналоги;

4. Зробити аналіз апаратного обладнання;

5. Виконати всі поставлені задачі випускної роботи;

6. Протестувати розробку та впровадити перед керівником  
підприємства;

7. Оформити результати у вигляді пояснювальної записки до  
відповідних ДСТУ норм випускних робіт молодшого спеціаліста.

Дата видачі завдання: 2023 року

#### 4. Календарний графік

№ етапу	Зміст	Терміни виконання	Готовність по графіку %, підпис керівника	Підпис керівника про повну готовність етапу, дата
1.	Збір практичного матеріалу за темою випускної роботи	16.01.23-17.02.23		
2.	<b>I атестація</b> I розділ випускної роботи молодшого спеціаліста (фахового молодшого бакалавра)	27.03.23-01.04.23		
3.	<b>II атестація</b> II розділ випускної роботи молодшого спеціаліста (фахового молодшого бакалавра)	01.05.23-06.05.23		
4.	<b>III атестація</b> III розділ випускної роботи молодшого спеціаліста (фахового молодшого бакалавра), висновки та рекомендації, додатки, реферат	29.05.23-03.06.23		
5.	Перевірка випускної роботи на оригінальність	15.05.23-12.06.23		
6.	Доопрацювання випускної роботи молодшого спеціаліста (фахового молодшого бакалавра), підготовка презентації, отримання відгуку керівника і рецензії	05.06.23-10.06.23		
7.	<b>Попередній захист випускної роботи молодшого спеціаліста (фахового молодшого бакалавра)</b>	12.06.23-18.06.23		
8.	Подача випускної роботи молодшого спеціаліста (фахового молодшого бакалавра) на кафедрі	за 3 дні до захисту		
9.	<b>Захист випускної роботи молодшого спеціаліста (фахового молодшого бакалавра)</b>	19.06.23-24.06.23		

Керівник Дерева Катерина Володимирівна (ПІБ) «    »      2023р.

Студент Бибик Тарас Ігорович (ПІБ) «    »      2023р.

## РЕФЕРАТ

Випускна робота молодшого спеціаліста включає: 69 сторінок, 21 рисунок, 4 лістинги, 13 інтернет-посилань, 1 додаток.

Об'єкт роботи: годинник-метеостанція під управлінням мікроконтролера ESP8266.

Метою роботи стало створення власного аналогу годинника-метеостанції під управлінням мікроконтролера ESP8266.

Основними задачами випускної роботи стали:

1. Ознайомлення з роботою метеостанцій та аналіз аналогів представлених на ринку;
2. Вибір програмного забезпечення та складових апаратної частини проекту;
3. Реалізація власного аналогу годинника-метеостанції.

В роботі вказано способи використання метеостанцій у різних галузях та продемонстровано процес розробки власного пристрою.

Результатом випускної роботи є представлений годинник-метеостанція, який може використовуватися для отримання даних про стан навколишнього середовища.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	7
ОГЛЯД ТА АНАЛІЗ МОЖЛИВОСТЕЙ МЕТЕОСТАНЦІЙ.....	11
1.1    Актуальність теми .....	11
1.2    Основні відомості про метеостанції .....	11
1.3    Домашні метеостанції.....	14
1.4    Огляд існуючих аналогів.....	16
1.4.1    La Crosse WS9057 .....	16
1.4.2    Netatmo Urban Weather Station .....	18
1.4.3    Beurer HM 16.....	20
1.5    Постановка завдання .....	21
1.6    Висновки за розділом .....	22
ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АПАРАТНИХ КОМПОНЕНТІВ .....	23
2.1    Вибір апаратної складової.....	23
2.1.1    LoLin NodeMCU V3 ESP8266 .....	23
2.1.2    MQ-135 .....	25
2.1.3    BMP280 .....	27
2.1.4    DHT22.....	28
2.1.5    DS3231.....	30
2.1.6    LCD1602.....	31

2.1.7	Плата розширення NodeMCU для ESP8266 .....	32
2.1.8	Матрична клавіатура .....	33
2.2	Вибір програмного забезпечення.....	34
2.2.1	Середовище програмування Arduino IDE .....	35
2.2.2	Вибір необхідних бібліотек.....	37
2.2.3	Fritzing .....	38
2.3	Висновки за розділом .....	39
ГОДИННИК МЕТЕОСТАНЦІЯ НА ESP8266.....		40
3.1	Перша версія годинника-метеостанції .....	40
3.1.1	Апаратне виконання .....	40
3.1.2	Програмне виконання.....	42
3.2	Зміна режимів .....	44
3.2.1	Зміна режимів .....	44
3.3	Функція актуалізації часу.....	46
3.4	Створення локального Web серверу .....	47
3.5	Тестування пристрою .....	49
3.6	Висновки за розділом .....	51
ВИСНОВКИ .....		53
ПЕРЕЛІК ПОСИЛАНЬ .....		54
ДОДАТКИ .....		56
ДОДАТОК А. ВИХІДНИЙ КОД ПРОГРАМИ.....		56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА  
ТЕРМІНІВ

Слово/Словосполучення	Скорочення
А	
Ампер	А
Б	
Байт	Б
В	
Вольт	V/В
Г	
Герц	Гц
М	
Міліметр	mm/мм
П	
Паскаль	Pa/Па
Програмне забезпечення	ПЗ
Ш	
Штука/штук	шт
А	
	AVR
Advanced RISC Machine	ARM
В	
Basic Input Output System	BIOS
С	
Градус за шкалою Цельсія	°C
G	

General Packet Radio Service	GPRS
H	
Hour	h
I	
Inter-Integrated Circuit	I2C
In-Plane Switching	IPS
Integrated Developer Environment	IDE
L	
Light-emitting Diode	LED
Liquid Crystal Display	LCD
M	
Minute	m
O	
Output	OUT
P	
Parts per million	PPM
Power	PWR
R	
Random Access Memory	RAM
Reduced Instruction Set Computer	RISC
Real Time Clock	RTC
S	
Static Random Access Memory	SRAM
Serial Peripheral Interface	SPI
System on Crystal	SoC
Second	s
T	



Thin Film Transistor	TFT
U	
Universal Asynchronous Receiver- Transmitter	UART
Universal Serial Bus	USB
W	
Wireless Fidelity	Wi-Fi

## ВСТУП

Останнім кроком, у навчальному процесі, студентів коледжу – є написання випускної роботи. Це обов'язковий етап, необхідний для завершення навчання. Студенти групи «Комп'ютерна інженерія» повинні створити апаратно-програмний продукт на обрану раніше тему.

На сьогоднішній день технології зробили стрімкий стрибок уперед. Зараз існує безліч різних вимірювальних приладів, які допомагають людям у найрізноманітніших сферах діяльності. Від вольтметрів до апаратів контролю стану здоров'я. У рамках даної випускної роботи, річ зайде про інші вимірювальні пристрої, а саме про метеостанції.

Метеостанції – це дуже корисна річ, у сучасному світі. Вона допомагає дізнатися прогноз погоди та показує поточні показники навколишнього середовища. Можливості метеостанцій на цьому не закінчуються, у різних сферах функціонал подібних пристроїв розширюють для додаткових конкретних задач.

## РОЗДІЛ 1

### ОГЛЯД ТА АНАЛІЗ МОЖЛИВОСТЕЙ МЕТЕОСТАНЦІЙ

#### 1.1 Актуальність теми

Майже кожна людина слідкує за своїм здоров'ям. Одним з факторів, що впливають на загальний стан людини – те як людина одягається. Для цього люди або дивляться у вікно, щоб дізнатися яка погода та що одягають інші, або ж звертаються до прогнозу погоди.

Жоден прогноз погоди не можливий без спеціальних метеостанцій, які знімають показники стану навколишнього середовища, повідомляють теперішній стан та прогнозують майбутній стан погоди. Подібні пристрої використовують не тільки метеорологи а й прості люди, проте розміри та можливості метеостанцій можуть кардинально відрізнятись.

Саме через необхідність дізнатися прогноз погоди, метеостанції набули широкого поширення, що закріпило відповідь на питання про їх актуальність. На поточний момент ще немає такого пристрою, який міг би повноцінно замінити метеостанції, адже вони вже довгий час допомагають людям слідкувати за змінами погоди, що корисно не тільки простим людям, які бажають вийти на вулицю, але й науковцям які проводять різні метеорологічні дослідження.

#### 1.2 Основні відомості про метеостанції

Метеостанція – це прилад, що в залежності від встановлених датчиків, виконує виміри температури, вологості тиску, тощо та виводить їх на екран або наприклад, на окремий сервер.

В залежності від сфери на яку розрахована певна метеостанція, цей пристрій може бути як компактним – домашні метеостанції, так і навпаки дуже масштабним як у метеорологічних центрах.

Існують як цифрові, так і аналогові метеорологічні станції.

На класичній (аналоговій) метеостанції розташовані такі датчики:

- термометр – вимірює температуру;
- барометр – вимірює тиск;
- гігрометр – вимірює вологість повітря;
- анемометр – вимірює швидкість вітру;
- флюгер – вимірює напрямок вітру;
- опадомір – вимірює кількість опадів.

Існують чотири основні типи метеостанцій:

1. Дорожні – окрім перерахованих вище датчиків, дані станції використовують датчик температури поверхні та датчик температури на глибині 30 см, контролер та GPRS модуль, щоб передавати отримані дані до інформаційних центрів. Для інформування водіїв можуть бути встановлені інформаційні табло з температурою поверхні та повітря, додатково можуть з'являтися попередження, наприклад, «мокра дорога» або «бічний вітер», тощо;
2. Лісові – вимірюють вологість дерева, ґрунту, температуру на різних рівнях висотності лісів, після чого отримані дані обробляються та моделюється карта пожежної активності, що й допомагає пожежникам запобігти можливих займань або поширенню пожежі;
3. Гідрологічні – такі станції ведуть спостереження над станом погоди над водоймами. Дані станції розташовують на материках, морських плаваючих станціях, а також у річках, озерах, болотах – відповідно;
4. Побутові або домашні – функціонал таких станцій дуже схожий з повноцінною метеорологічною станцією, проте обробляє така станція менше даних, що надходять від одного або декількох датчиків розташованих за вікном та в інших приміщеннях. Домашні метеостанції показують температуру в та поза приміщенням, вимірюють вологість, атмосферний тиск та виходячи з отриманих даних формують прогноз погоди на добу. Існують як портативні (зі

змінним джерелом живлення) так і стаціонарні (що працюють від електромережі) домашні метеостанції.

Метеостанції використовуються у різних сферах діяльності:

1. Сільське господарство. Польові метеостанції використовують аграрії, щоб створити оптимальні умови для росту культур. Польові метеостанції складаються з двох блоків:

- блок планування завдань – використовують для контролю поливу, внесення добрив, обробки ґрунту та збирання врожаю
- блок аналізу ефективності виробництва.

Основні дані які отримують дані метеостанції – дані про опади, температуру та вологість повітря, атмосферний тиск, швидкість та напрямок вітру, температуру та вологість ґрунту, сонячну активність, вологість та температуру листа (для садів).

2. Воєнна справа. Метеостанції-вітроміри використовують стрільці для корегування балістики патрону. Дані пристрої це покращена версія кишенькового анемометру. Такі метеостанції дуже компактні, що дозволяє носити їх у кишені та дозволяють отримати дані необхідні для корегування пострілу.

Метеостанції-вітроміри вимірюють такі показники: барометричний тиск, висота, бічний вітер, висота по щільності, температура точки роси, зустрічний вітер, індекс спеки, відносна вологість повітря, тиск на рівні метеостанції (абсолютний тиск), температура, температура за вологим термометром, температура з урахуванням охолодження вітром, швидкість та напрямок вітру.

Найчастіше використовують такі пристрої для корегування пострілів на далеку та наддалеку дистанції.

Такі метеостанції є ударостійкими та живляться від батарейок/акумуляторів.

### 1.3 Домашні метеостанції

Домашні метеостанції – це пристрої, які використовують коли необхідно слідкувати за станом повітря у квартирі або щоб дізнатися про погоду на вулиці у даний момент.

Більшість метеостанцій вимірюють стандартні показники повітря, проте багато сучасних моделей оснащені також додатковими функціями:

- виводять інформацію про погодинну/щомісячну/щорічну кількість опадів;
- швидкість вітру;
- визначення показників ультрафіолетового випромінювання та ін.

Через велику кількість варіантів домашніх метеостанцій, наданий момент існує багато різних варіацій з різними можливостями та характеристиками:

1. Дисплей. Метеостанції можуть мати як маленький та і великий екран, на якому можна вивести певну інформацію. Усе залежить від виробника, наприклад, екран маленький, проте на ньому чітко видно інформацію, але потрібно перемикатися між станами інформаційної панелі, щоб переглянути різну інформацію або навпаки – великий екран з чітким зображенням та усією необхідною інформацією, що виключає необхідність перемикання інформаційної панелі.
2. Віддалений доступ. В деяких моделях метеостанцій є можливість надсилання інформації на віддалений сервер, що дозволяє користувачу подивитися усі необхідні дані, наприклад, через сайт виробника або окремий додаток у смартфоні.
3. Діапазон датчиків. В залежності від цінового діапазону у домашніх метеостанціях використовуються різні датчики, саме тому вони відрізняються діапазоном дії. В середньому діапазон дії складає 30 метрів, проте існують рішення в яких цей параметр складає 100 або навіть близько 300 метрів.

4. Джерело живлення. На даний момент існує три варіанти джерел живлення у подібних метеостанціях:

- живлення від вмонтованого у метеостанцію акумулятора;
- живлення від батарейок;
- живлення від блоку на 220В.

Перші 2 варіанти дозволяють зробити метеостанції більш портативними, а також виключити залежність роботи пристрою від електроенергії, що дозволяє користуватися станцією у будь-який момент часу, головне слідкувати за зарядом. Чого не можна сказати про варіант з блоком живлення на 220В, адже якщо не буде світла, станція не зможе працювати.

5. Точність вимірювання показників. Різні метеостанції відрізняються різною точністю вимірювання. На точність вимірювань можуть впливати декілька факторів:

- безпосередньо, сам датчик, що використовується у конкретному пристрої;
- матеріал з якого виконана метеостанція;
- конструкція метеостанції;
- місце, в якому розміщені датчики та метеостанція;
- частота вимірювання (раз на хвилину/10хвилин/годину, тощо).

6. Довговічність. Це одна з найважливіших особливостей домашніх метеостанцій. Від того наскільки довговічні датчики та сама метеостанція залежить як довго буде працювати станція. Також довговічність впливає на точність показників, адже не виключено що датчик може бути лише частково справним, тобто він буде виконувати заміри, проте точність може не відповідати заявленим характеристикам. Саме тому датчики повинні бути достатньо «витривалі», щоб працювати при різних погодних умовах.

## 1.4 Огляд існуючих аналогів

Щоб краще зрозуміти, які компоненти необхідні для створення годиннику-метеостанції та як працюють домашні метеостанції, необхідно переглянути декілька аналогів представлених на ринку.

### 1.4.1 La Crosse WS9057

La Crosse – це французька компанія, що випускає у продаж домашні метеостанції, годинники, датчики, тощо. На даний момент компанія пропонує на ринку близько 10 продуктів. Компанія займається випуском як професійних так і домашніх пристроїв.

За даними виробника домашня метеостанція La Crosse WS9057 характеризується високою точністю вимірювань, а також за словами компанії La Crosse, дана метеостанція має різні унікальні особливості, що не властиві для подібної продукції від інших виробників.

La Crosse WS9057 на вигляд схожа на цифрову рамку, завдяки чому конструкція метеостанції дозволяє повісити її на стіну або просто поставити, наприклад, на столі, для цього з тильного боку пристрою є кріплення та висувні підставки.

Через конструкцію метеостанції панель керування розташована з тильної сторони пристрою. Усього на панелі керування розташовано 5 кнопок, які дозволяють:

- відрегулювати базові параметри;
- подивитися значення датчиків;
- підібрати канал для підключення бездротових датчиків.

Використання метеостанції у темну пору доби може бути утруднено через відсутність функції підсвічування у метеостанції.



Живлення станції відбувається від двох батарейок формату R14. Для скидання налаштувань метеостанції, необхідно на деякий час витягнути батарейки та повернути їх на місце, аналогічно до способу скидання BIOS у більшості материнських плат, в яких відсутня така кнопка.

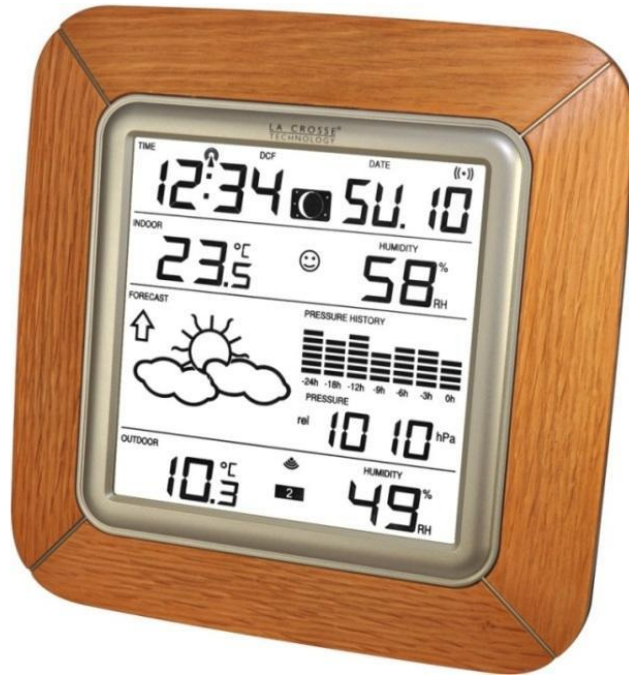


Рисунок 1.1 – Зовнішній вигляд La Crosse WS9057

Переваги La Crosse WS9057:

- дуже висока точність вимірювань;
- велика діагональ рідкокристалічного монохромного екрану;
- якість вихідного продукту.

Недоліки La Crosse WS9057:

- відсутність підсвічування;
- відсутність кнопки «Повернення до заводських налаштувань».

#### 1.4.2 Netatmo Urban Weather Station

Дана метеостанція також створена французькими ученими. Пристрій підтримує дуже широкий функціонал.

Комплектація даного продукту:

- станція;
- виносний модуль;
- адаптер для підключення до мережі;
- батарейки;
- кріплення для зовнішньої частини станції.

Внутрішній та зовнішній елементи пристрою виконані у вигляді циліндру, корпус виготовлений з якісного алюмінію. Має пластикову підставку.

Перегляд інформації можливий лише на окремий пристрій (ноутбук, комп'ютер, телефон, планшет), щоб отримати доступ до інформації необхідно встановити відповідний додаток, попередньо пройшовши обов'язкову процедуру реєстрації.

Перед використанням пристрою, при першому ввімкненні потрібно налаштувати WI-FI модуль, щоб метеостанція мала змогу зв'язатися з усіма зовнішніми пристроями, включаючи виносний блок. Вказана метеостанція може працювати лише з однією мережею, тобто у разі потреби зміни мережі доведеться знову пройти процедуру налаштування.

Станція відрізняється великим набором датчиків та відповідних налаштувань. Спектр функціоналу пристрою:

- стандартний набір даних про погоду за вікном;
- дані про концентрацію вуглекислого газу у повітрі;
- дані про шумове забруднення як на вулиці, так і у приміщенні.

Усі отримані дані зберігаються на окремому сервері, тому є можливість перегляду даних у конкретний момент часу.

Додаток для користування станцією простий у налаштуванні та має зрозумілий інтерфейс. У додатку можна переглянути точний прогноз погоди на найближчі декілька днів.

Модуль бездротового зв'язку відрізняється високою потужністю, що дозволяє розташувати блоки метеостанції на значній відстані один від одного. Внутрішній модуль станції допоможе визначити комфортність перебування у приміщенні та надасть інформацію, що необхідно зробити для досягнення ідеальних параметрів.



Рисунок 1.2 – Зовнішній вигляд Netatmo Urban Weather Station

#### Переваги Netatmo Urban Weather Station:

- широкий функціонал;
- велика кількість високоточних датчиків;
- можливість підключення додаткових блоків;
- наявність модуль концентрації вуглекислого газу.

#### Недоліки Netatmo Urban Weather Station:

- відсутність власного дисплею;
- висока вартість продукту.

### 1.4.3 Beurer HM 16

Метеостанція Beurer HM 16 – це компактний пристрій для вимірювання показників температури та вологості повітря.

Конструкція пристрою дозволяє встановити його як на столі, так і повісити на стіну. Станція має великий дисплей на якому чітко видно інформацію написану крупним шрифтом. На корпусі встановлені дві кнопки для регулювання шкали температури.

Живлення пристрою відбувається від двох батарейок типу CR2025, що поставляються разом із пристроєм.

Функціонал станції дозволяє подивитися температуру та вологість навколишнього повітря та має можливість переглянути час.

Пристрій зберігає лише екстремальні показники датчиків, тобто максимальні та мінімальні значення, отримані в процесі роботи. Дані виводяться в двох температурних шкалах – градуси за Фаренгейтом або градуси за шкалою Цельсія.

Пристрій зазвичай використовується для контролю мікроклімату у приміщенні. Невеликі габарити дозволяють носити пристрій у кишені. При необхідності пристрій можна використовувати під час відпочинку на природі.



Рисунок 1.3 – Зовнішній вигляд Beurer HM 16

#### Переваги Veurer NM 16:

- простота використання;
- низька ціна;
- надійність та довговічність;
- якісна збірка;
- зручний інтерфейс.

#### Недоліки Veurer NM 16:

- відсутність підсвічування;
- скупий функціонал.

### 1.5 Постановка завдання

Щоб вдало реалізувати обраний проект необхідно визначити основні задачі для виконання, таким чином буде одразу зазначений план робіт який буде підказувати, що саме необхідно зробити.

Основні задачі під час роботи над випускною роботою:

- обрати усі компоненти, необхідні для створення апаратної частини проекту;
- обрати середовище для програмування мікроконтролерів та бібліотеки, необхідні для роботи обраних компонентів;
- під'єднати усі компоненти до мікроконтролеру;
- розробити код програми для роботи мікроконтролеру;
- завантажити створену програму до пам'яті мікроконтролеру;
- протестувати запрограмований пристрій.

## 1.6 Висновки за розділом

Під час написання першого розділу було розглянуто вже існуючі аналоги метеостанцій. Після більш детального ознайомлення з принципами роботи домашніх метеостанцій було встановлено їх переваги та недоліки. Також в ході написання першого розділу було поставлено певні задачі – необхідні до виконання.

Інформація отримана під час створення першого розділу випускної роботи доводить доцільність створення власного аналогу домашньої метеостанції, а саме програмно – апаратного комплексу для моніторингу стану навколишнього середовища з функцією «годинник» на базі мікроконтролера.

## РОЗДІЛ 2

### ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АПАРАТНИХ КОМПОНЕНТІВ

Основним завданням при виконанні випускної роботи студента спеціальності «Комп'ютерна інженерія» - є правильний вибір сумісних компонентів та програмного забезпечення, який вплине на подальшу роботу з проектом.

#### 2.1 Вибір апаратної складової

Першим етапом виконання практичної реалізації проекту став вибір мікроконтролеру на якому буде базуватися увесь проект, та компонентів на які будуть відповідати за функціонал годинника-метеостанції.

Для створення годинника-метеостанції знадобиться:

- мікроконтролер, для керування усіма компонентами;
- датчик вологості;
- датчик температури;
- барометр;
- датчик якості повітря;
- дисплей;
- модуль часу;
- плата розширення
- матрична клавіатура.

##### 2.1.1 LoLin NodeMCU V3 ESP8266

Щоб реалізувати проект випускної роботи першим що потрібно зробити – обрати мікроконтролер, який буде контролювати усі процеси.

Двома найрозповсюдженішими мікроконтролерами, якими користуються майже кожна людина, якій необхідно розробити подібний проект, це Arduino UNO R3 CH340 на базі ATmega328P та ESP8266.

Основні відмінності даних мікроконтролерів у тому, що ESP8266 має більший обсяг SRAM, а також модуль Wi-Fi. Також на відміну від Arduino, даний мікроконтролер особливий тим, що не має власної flash-пам'яті в SoC. Прошивка (завантаження файлу з кодом програми до пам'яті мікроконтролеру) ESP8266 відбувається завдяки зовнішній flash-пам'яті з інтерфейсом SPI.

На сьогоднішній день ESP8266 базується на двох USB конвертерах, CP2102 та CH340.

Саме через вже вказані причини, для виконання теми випускної роботи було обрано мікроконтролер LoLin NodeMCU V3 ESP8266 (CH340).

ESP8266 – даний мікроконтролер створений китайською компанією Espressif Systems у 2014 році.

Технічні характеристики ESP8266:

Напруга живлення	від 2,5 до 3,6 В
Вхідна напруга	від 4,5 до 10 В
Енергоспоживання	до 215 мА
Флеш-пам'ять	4 МБ
SRAM	64 КБ
Тактова частота	80 МГц
Цифрові виходи	11
Аналогові виходи	1
Інтерфейси	UART/SPI/I2C
Wi-Fi стандарт	802.11 b/g/n
USB порт	Micro USB





Рисунок 2.1 – LoLin NodeMCU V3 ESP8266 CH340

### 2.1.2 MQ-135

MQ-135 - це один з датчиків серії MQ, що випускаються для контролю якості повітря. Цей датчик використовується у більшості проектів у яких необхідно дізнатися рівень вуглекислого газу у повітрі. Замість наведеного датчику також використовують інфрачервоний датчик MH-Z19.

Датчик MQ-135 чутливий до таких газів та випарів:

- аміак ( $\text{NH}_3$ );
- оксид азоту ( $\text{NO}_x$ );
- етиловий спирт;
- бензол;
- дим;
- вуглекислий газ ( $\text{CO}_2$ ), тощо.

Датчик має два виходи – цифровий та аналоговий. Аналоговий вихід відповідає за надсилання значень PPM з датчику, цифровий – за передачу сигналу HIGH/LOW. Поріг спрацьовування датчику на відповідний рівень PPM налаштовується потенціометром на задній частині датчику.

Також датчик має 2 світлодіоди стану: PWR LED та OUT LED. Перший відповідає за ідентифікацію подачі напруги на датчик, другий – показує працює компаратор чи ні.

Основний мінус датчику полягає у тому, що перед використанням датчик необхідно прогрівати на протязі доби, щоб уникнути неточності вимірювання.

Технічні характеристики MQ-135:

Компаратор	LM393
Споживаний струм	до 180 мА
Потужність нагрівача	До 900 мВт
Аналоговий вихід датчику	0-5В. Чим більша концентрація, тим більша напруга на виході
Робоча температура, °С	-10 / +50
Напруга живлення модулю	5В
Розміри	32мм x 22мм x 27мм

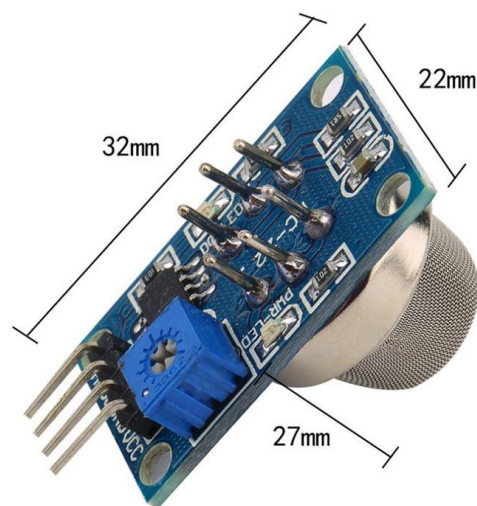


Рисунок 2.2 – Датчик якості повітря MQ-135

### 2.1.3 BMP280

Для вимірювання атмосферного тиску використовуються або BMP280 або BME280. Відмінність датчиків у тому, що BME280 дозволяє вимірювати вологість, проте ця функція робить BMP280 у 4 рази доступнішим, якщо говорити про вартість модулів. Тому саме BMP280 буде використовуватись у даному проекті.

BMP280 – це модуль вимірювання атмосферного тиску від фірми BOSCH. Цей датчик – покращена версія BMP180. Відмінності BMP280:

- менші розміри;
- зменшене енергоспоживання;
- вища точність;
- наявність точного калібрування з заводу;
- наявність двох послідовних інтерфейсів: I2C та SPI.

Модуль підтримує три режими роботи:

SLEEP – режим зменшеного енергоспоживання;

FORCED – за командою виходить з режиму сну, виконує заміри, передає дані контролеру та переходить у режим SLEEP;

NORMAL – датчик самостійно виходить з режиму сну, виконує заміри та переходить у режим SLEEP. Зчитування даних в такому режимі доступне у будь-який момент часу.

Датчик має можливість фільтрувати результати вимірювання:

- OVERSAMPLING для температури (16,17,18,19,20 біт);
- OVERSAMPLING для тиску (16,17,18,19,20 біт);
- TSB – час між замірами (0.5, 62.5, 125, 250, 500, 1000, 2000, 4000 мс);
- FILTER\_COEFFICIENT – коефіцієнт фільтрації.

## Технічні характеристики BMP280:

Напруга живлення	від 1,7В до 3,6В
Максимальна швидкість інтерфейсу I2C	3,4МГц
Споживаний струм	2,7мкА при частоті 1 Гц
Інтерфейси	I2C, SPI (4 дроти), SPI (3 дроти)
Калібрування	з заводу
Діапазон вимірювання тиску	від 300 гПа до 1100 гПа
Розміри	21мм x 18мм

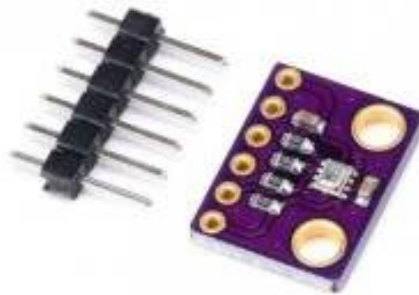


Рисунок 2.3 – Датчик тиску BMP280

## 2.1.4 DHT22

DHT22 – це датчик температури та вологості. В конструкції датчику використовуються ємнісний датчик температури та гігрометр. Дані з датчику передаються за рахунок цифрових сигналів. Перетворення значень датчику до цифрових відбувається завдяки внутрішньому аналого-цифровому перетворювачу.

DHT22 є більш дорогим та більш точним аналогом датчику DHT11. Основна відмінність датчиків у тому, що DHT11 не може вимірювати значення температури нижче 0 °С та вище 50 °С, а також має менший діапазон вимірювання значень вологості – від 20 до 90%.

Датчик має 4 піни підключення, 3 з яких необхідні для роботи. Для підключення до датчику потрібно під'єднати підтягуючий резистор на 10кОм. Проте датчики DHT11 та DHT22 бувають двох видів: без обв'язки, тобто просто датчик з 4-ма виходами та з обв'язкою, тобто датчик вмонтований на текстолітову плату з 1-м або 2-ма резисторами, в залежності від виробника, та має 3 виходи, необхідні для роботи.

#### Технічні характеристики DHT11:

Напруга живлення, В	3,3 – 6
Діапазон вимірювання вологості, %	0 – 100%
Діапазон вимірювання температури, °С	-40 / + 80
Похибка вимірювання вологості, %	± 2
Похибка вимірювання температури, %	± 0,5
Частота опитування, Гц	До 0,5
Споживаний струм	до 2,5 мА

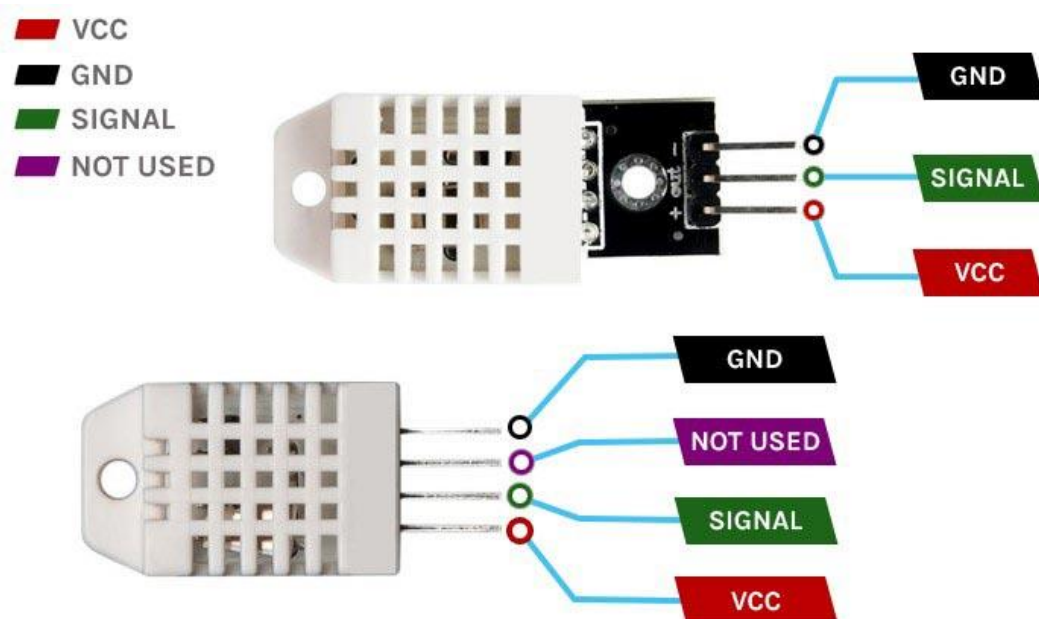


Рисунок 2.4 – Два варіанти виконання конструкції датчику DHT11.

### 2.1.5 DS3231

Враховуючи що наша метеостанція також повинна виконувати функції годинника, необхідно додати модуль реального часу. Існує декілька моделей RTC модулю: DS1302, DS1307 та DS3231.

Для реалізації функції годинника було обрано модуль DS3231. Даний модуль створений на основі мікросхеми DS3231N та мікросхеми пам'яті AT24C32N. Мікросхема DS3231N складається з таких компонентів:

- кварцовий резонатор;
- датчик температури;
- схема термічної компенсації.

Мінімальна одиниця часу в даній мікросхемі – секунда, максимальна – рік, таким чином обраний модуль може показувати максимально точні дату та час. Також модуль робить поправку на високосний рік та підтримує 2 формати часу: 12-ти та 24-х годинний. Підключається модуль за допомогою шини I2C.

Технічні характеристики DS3231:

Напруга живлення	3-5,5В
Інтерфейс	I2C
Робоча температура	-40/+85 °C
Розміри модулю, мм	38 x 22 x 14



Рисунок 2.5 – Модуль реального часу DS3231

### 2.1.6 LCD1602

Останнє про що потрібно подбати при створенні годинника метеостанції – це зручність використання пристрою. Щоб користувач зміг повноцінно використовувати наш прилад необхідно дати йому доступ до інформації, а саме дозволити йому переглядати усю необхідну інформацію про стан навколишнього середовища та дати/часу. Для цього необхідно обрати дисплей для виводу інформації.

Найчастіше в подібних проектах використовують LCD та TFT IPS дисплеї (далі TFT). Основна відмінність таких дисплеїв це кольорова IPS матриця, за рахунок чого можна робити більш зручний інтерфейс. Проте робота з TFT дисплеями складніша ніж з LCD, а також TFT дисплей повинен бути великого розміру, для того аби користувач чітко бачив усю інформацію, що робить такі дисплеї менш доступними. Саме тому для випускної роботи був обраний LCD дисплей.

Серед LCD дисплеїв, найрозповсюдженіші це LCD 1602 та LCD 2004. Хоча для обраної теми краще підійшов би LCD 2004, проте на момент виконання роботи доступний лише LCD 1602.

Назва дисплею чітко вказує на кількість стовпців та рядків дисплею, а саме 16 стовпців на 2 рядки. Даний дисплей виконаний на базі контролеру HD44780. Дисплей підтримує два режими роботи: 4-бітний та 8-бітний. Різниця між режимами роботи – підтримка старших бітів у 8-бітному режимі, проте для цього потрібно додатково підключити ще 4 контакти.

Існує декілька варіантів виконання даного дисплею з I2C модулем або без нього. В даному випадку використовується LCD дисплей з I2C модулем. Наявність вище вказаного модулю спрощує підключення за рахунок усього 4-х контактів для підключення, а також дозволяє користуватися доступом через шину I2C.

## Технічні характеристики LCD1602:

Напруга живлення, В	5
Робоча температура, °С	-20 / +70
Колір дисплею	Синій
Колір символів	Білий
Формат	16 x 2
Видима область, мм	66 x 16
Контролер	HD44780
Інтерфейс	I2C

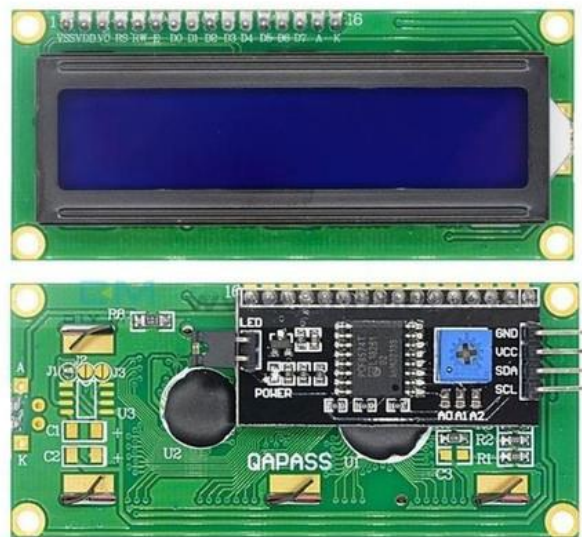


Рисунок 2.6 – Дисплей LCD 1602

## 2.1.7 Плата розширення NodeMCU для ESP8266

Версія ESP8266, яка використовується у даному проєкті, не підходить для стандартного використання разом з макетною платою через фізичні розміри плати мікроконтролера. Дана ESP8266 займає усі місця для підключення датчиків, залишаючи лише місця для підключення ліній 5В та GND, тобто ширина плати мікроконтролера займає усе місце, необхідне для



встановлення датчиків або дротів на макетну плату. Також у ESP8266 бракує виходів для підключення усіх необхідних датчиків.

Щоб вирішити вище описані проблеми було обрано плату розширення яка дозволяє підключити набагато більшу кількість модулів до плати мікроконтролеру. Плата розширення має чотири виходи на один цифровий вихід ESP, тобто на один вихід, наприклад D1, є можливість підключити до 4-х пристроїв. Логіка мікроконтролеру дозволяє розрізняти різні пристрої на одному виході, тобто проблем в роботі виникнути не повинно.

Обрана плата розширення має виходи 5В та 3,3В для живлення датчиків, LED - індикатор живлення, DC-DC понижуючий перетворювач постійного струму 5В/1А а також гніздо 5,5 x 2,1мм для живлення плати.

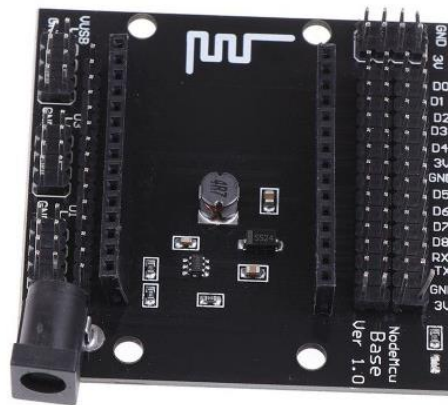


Рисунок 2.7 – Плата розширення NodeMCU

### 2.1.8 Матрична клавіатура

Щоб користувач міг давати команди пристрою, необхідно щоб були відповідні клавіші. Для цього було обрано матричну клавіатуру розміром 4x4. За допомогою вказаної клавіатури з'являється можливість розширити існуючий функціонал, а також зменшується кількість компонентів, необхідних для роботи пристрою.

Усього доступно 16 клавіш, які можна запрограмувати необхідним чином. Реакція на натискання клавіш відбувається за рахунок замикання контактів, між рядками та стовпцями, у відповідних місцях. Для підключення використовується шлейф на 8 виходів, по 4 на рядки та стовпці.

Основний мінус клавіатури – хибні спрацьовування, при натисканні можуть спрацювати інші клавіші.

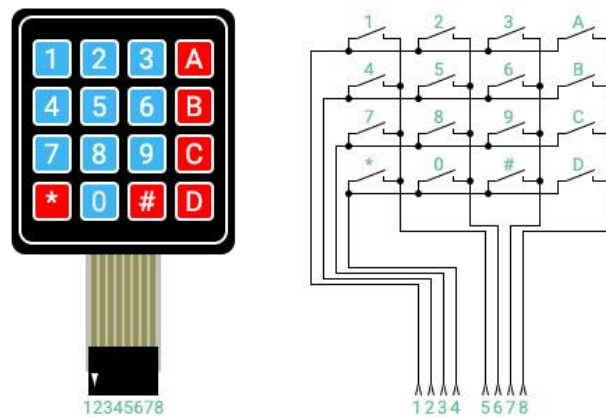


Рисунок 2.8 – Матрична клавіатура

## 2.2 Вибір програмного забезпечення

Після того як було обрано усе необхідне для апаратної частини проекту, потрібно визначитися яке програмне забезпечення буде використовуватися для написання коду та моделювання схем підключення.

На момент виконання випускної роботи, єдиним редактором коду, з можливістю завантаження скетчу (файлу з кодом програми) до пам'яті мікроконтролера, який раніше використовувався для реалізації подібних проектів – Arduino IDE.

Для реалізації проекту також необхідно обрати список бібліотек, що будуть використовуватися під час написання коду.

### 2.2.1 Середовище програмування Arduino IDE

Arduino IDE – це інтегроване середовище розробки, створене компанією Arduino Software. На сьогоднішній день, це найпоширеніше середовище розробки для мікроконтролерів AVR, ARM, ESP та Intel x86.

Альтернативними рішеннями, що використовуються для створення подібних проектів, можна назвати:

- AVR Studio (наразі інтегрована у програму Atmel Studio);
- Visual Studio Code (якщо завантажити модуль PlatformIO);
- Eclipse.

Перевагою, над вище вказаними аналогами, є те, що Arduino IDE має досить простий та зручний у використанні інтерфейс.

Основне вікно програми виглядає таким чином:

- 1) Головне меню програми – поєднує у собі увесь функціонал програми, тобто кожна дія (окрім безпосереднього написання коду у файлі), що доступна до виконання програмою, може бути виконана через головне меню.
- 2) Меню гарячих клавіш – в даному меню розташовано 6 основних функцій програми, що використовуються найчастіше (зліва - направо):
  - перевірити скетч на наявність помилок;
  - завантажити скетч до пам'яті мікроконтролеру
  - створити новий файл;
  - відкрити раніше збережений файл;
  - зберегти файл.
- 3) Поле з усіма файлами проекту;
- 4) Поле текстового редактору – в даному полі відбувається написання коду програми;

- 5) Поле для службових повідомлень – може демонструвати хід перевірки програми, повідомляти про успішне/невдале завершення процесу завантаження програми, тощо;
- 6) Вікно в якому відображається інформація про помилки/попередження, хід компіляції коду, розмір файлу у пам'яті мікроконтролеру, прогрес завантаження файлу у пам'ять мікроконтролеру, тощо.

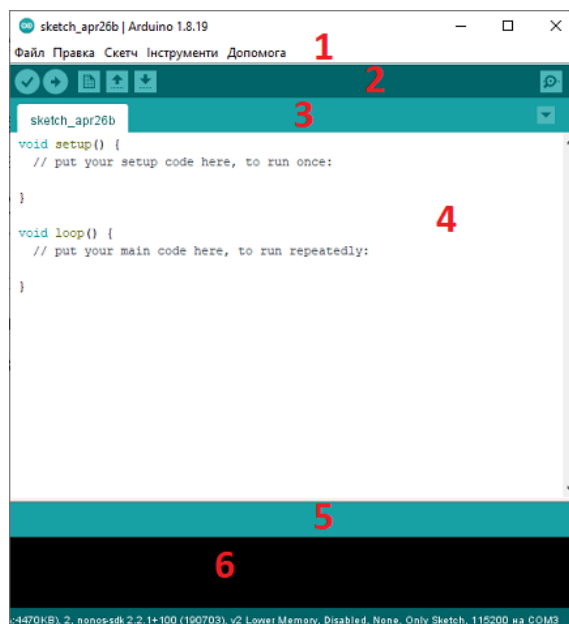


Рисунок 2.9 – Інтерфейс Arduino IDE

Завантажити Arduino IDE можна на сайті, за адресою: <https://www.arduino.cc/en/software>, або на відкритому GitHub репозиторію: <https://github.com/arduino/Arduino>.

Під час виконання випускної роботи використовувалася Arduino IDE версії 1.8.19, це остання 1.8.X версія, що стабільно працює при роботі з мікроконтролерами. На момент написання другого розділу, Arduino IDE було оновлено до останньої версії 2.1.0.

## 2.2.2 Вибір необхідних бібліотек

Стандартного переліку бібліотек, наданих Arduino IDE, недостатньо для даного проекту, тому для початку потрібно встановити усі необхідні бібліотеки через менеджер бібліотек. Для цього, потрібно, скориставшись головним меню програми, перейти до вкладки Інструменти → Керування бібліотеками, або натиснути комбінацію клавіш Ctrl + Shift + I.

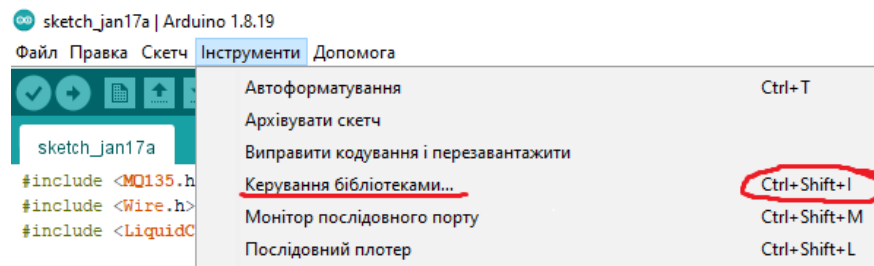


Рисунок 2.10 – Спосіб переходу до менеджера бібліотек.

Після проведених маніпуляцій повинно з'явитися вікно зі списком усіх бібліотек, доступних для Arduino IDE.

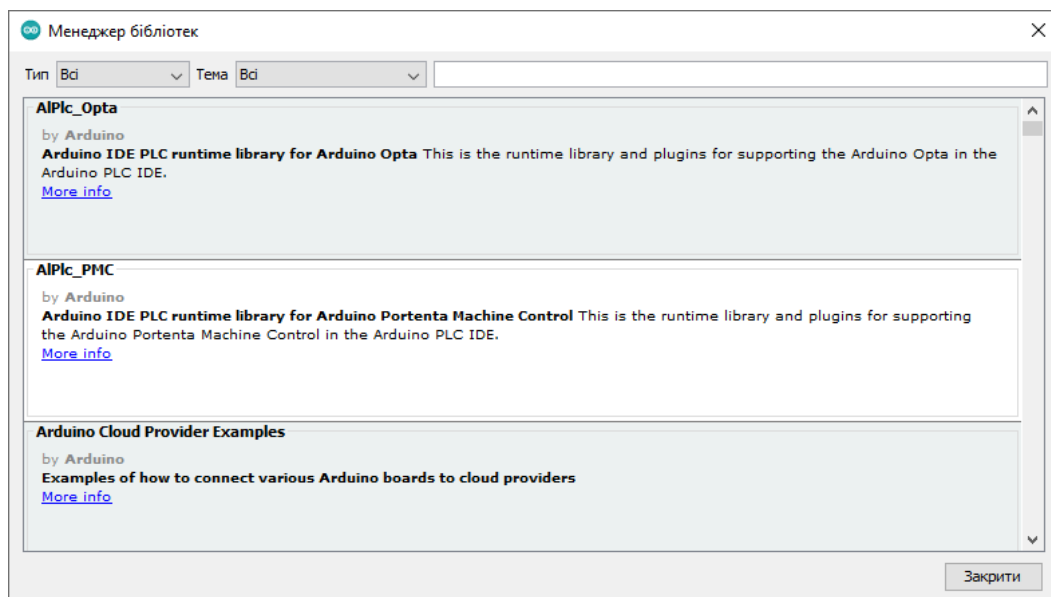


Рисунок 2.11 – Вікно менеджера бібліотек

Менеджер бібліотек дозволяє відсортувати список бібліотек та знайти потрібну бібліотеку через пошук. Якщо необхідної бібліотеки не знайдено, її можна завантажити через Інтернет.

Перелік необхідних бібліотек:

- Adafruit BMP280 Library;
- DHT sensor library;
- LiquidCrystal I2C;
- MQ135;
- RTCLib;
- Wire;
- Keypad.

### 2.2.3 Fritzing

Для моделювання схем підключення було вирішено обрати програму Fritzing. Fritzing поєднує усе необхідне для роботи з мікроконтролерами:

- середовище для моделювання схем підключення на монтажній платі;
- середовище для моделювання схем електричних принципів;
- середовище для моделювання друкованих схем;
- редактор деталей;
- середовище програмування.

Усі компоненти для моделювання можна знайти у меню «Набір деталей». За необхідністю є можливість завантажити бібліотеки з відсутніми модулями.

У вікні програмування є серійний монітор та можливість завантаження коду до пам'яті мікроконтролера (аналогічно до Arduino IDE).

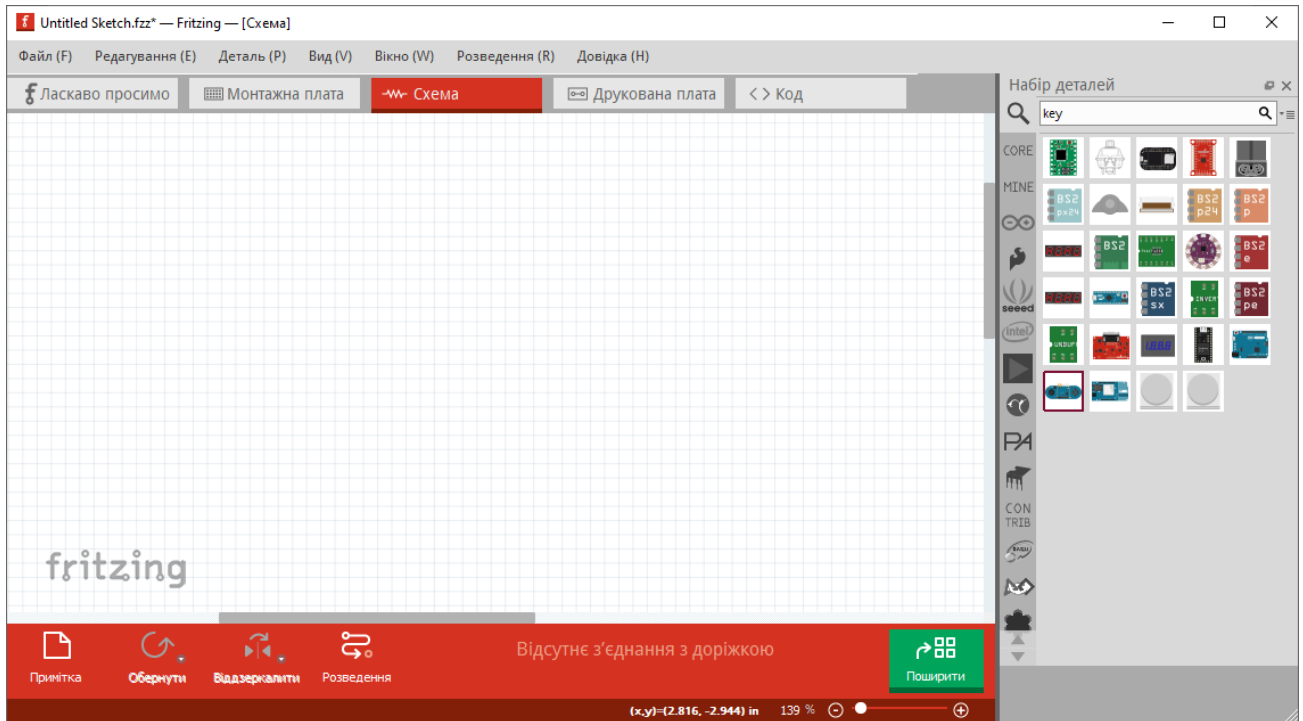


Рисунок 2.12 – Вікно програми Fritzing

### 2.3 Висновки за розділом

Під час роботи над другим розділом було оглянуто найуживаніше програмне забезпечення для програмування та моделювання схем, а також обрано доступні, але достатньо точні та практичні модулі, для створення годиннику-метеостанції.

Під перелік обраних компонентів було підібрано бібліотеки, необхідні для їх правильного функціонування.

## РОЗДІЛ 3

### ГОДИННИК МЕТЕОСТАНЦІЯ НА ESP8266

Щоб під час процесу розробки не виникало проблем вирішено працювати поетапно, тобто додавання функціоналу буде відбуватися поступово та зі збереженням попередніх версій, для можливості відкату у разі виникнення неполадок.

#### 3.1 Перша версія годинника-метеостанції

##### 3.1.1 Апаратне виконання

Перша версія годинника-метеостанції та її подальші модифікації з боку апаратної частини будуть виконуватися за допомогою макетної плати та плати розширення. Основними причинами, що завадять зробити пристрій більш компактним та з використанням паяльних засобів, стали:

1. Відсутність навичок пайки;
2. Відсутність паяльних засобів;
3. Можливість модульної заміни компонентів;
4. Наявність виходів 5V.

Для першої версії пристрою необхідно:

- макетна плата;
- плата розширення;
- ESP8266;
- DHT11;
- MQ-135;
- DS3231;
- LCD1602 I2C;



- BMP280;
- резистор з опором в 10 кОм;
- дріт типу «мама - мама» - 15 шт;
- дріт типу «мама - тато» - 3 шт;
- дріт типу «тато - тато» - 2 шт.

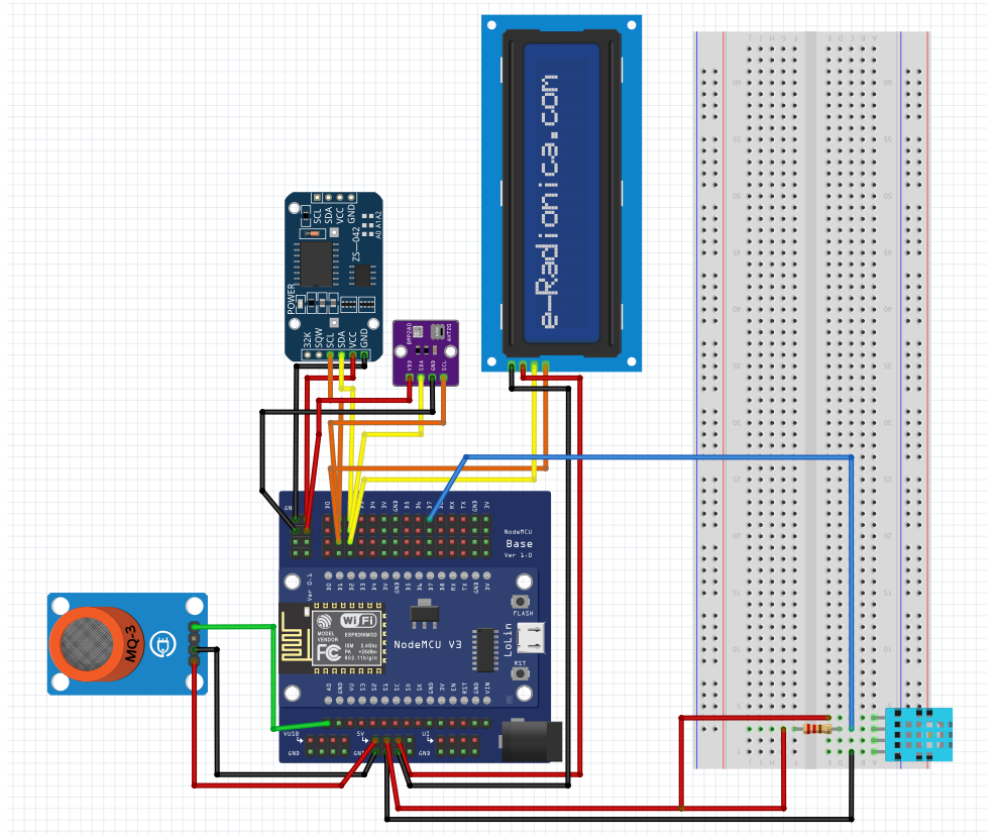


Рисунок 3.1 – Схема першої версії пристрою

ПРИМІТКА! Кінцеве виконання та надані, на першому та наступних етапах, схеми підключення можуть відрізнятися. Схеми створені для максимально простої демонстрації підключення модулів у єдиний функціональний проект.

### 3.1.2 Програмне виконання

В першому етапі проекту, функціонал доволі простий та не потребує особливих навичок у програмуванні. ESP8266 отримує дані з датчиків та відображає їх на екрані. Робота з кожним датчиком відбувається по черзі:

Отримав дані → Вивів інформацію на екран → Зачекав 3 секунди → Перейшов до наступного модулю.

Таким чином ESP8266 встигає отримати дані та дає користувачу можливість побачити усю необхідну інформацію. Перед кожним виводом даних на екран, попередні дані витираються з екрану, це дозволяє уникнути зайвих символів, які засмічують зображення та роблять його менш придатним до читання.

Отже усього в коді чотири блоки для роботи з відповідним модулем:

- для MQ-135;
- для BMP280;
- для DHT11;
- для DS3231.

Ці блоки розділені командами, що відповідають за затримку у 3 секунди та за очищення екрану:

```
delay(3000);
```

```
LCD.clear();
```

Лістинг 3.1 – Отримання та виведення даних з BMP280:

```
LCD.setCursor(0,0);
```

```
LCD.print("Temp = ");
```

```
LCD.print(bmp.readTemperature());
```

```
LCD.print(" *C");
```

```
LCD.setCursor(0,1);
```

```
LCD.print("P = ");
```

```
LCD.print bmp.readPressure();  
LCD.print(" Pa");
```

Усі блоки коду мають вказану вище структуру, за винятком блоку з RTC модулем. Цей блок необхідно модифікувати таким чином щоб на екрані правильно відображався час.

Бібліотека RTCLib працює таким чином, що дні, місяці, роки, секунди, хвилини та години зберігаються в окремих змінних, по одній на кожен міру часу. Тому коли години, хвилини, та секунди мають значення менше 10 на екрані з'являються зайві символи, тобто формат часу схожий на h:m:s. Через це, коли значення часу менше 10, необхідно виводити "0", щоб формат часу завжди був hh:mm:ss, тоді на екрані завжди буде однакова кількість символів і користувач буде бачити час у стандартному форматі.

Лістинг 3.2 – Виведення часу у стандартному hh:mm:ss форматі:

```
if (now.hour() < 10) LCD.print("0");  
    LCD.print(now.hour(), DEC);  
    LCD.print(':');  
if (now.minute() < 10) LCD.print("0");  
    LCD.print(now.minute(), DEC);  
    LCD.print(':');  
if (now.second() < 10) LCD.print("0");  
    LCD.print(now.second(), DEC);
```

Слід зауважити, що датчик MQ-135, при кожному запуску, повинен спочатку трохи прогрітися, тому деякий час показник якості повітря буде неточним.

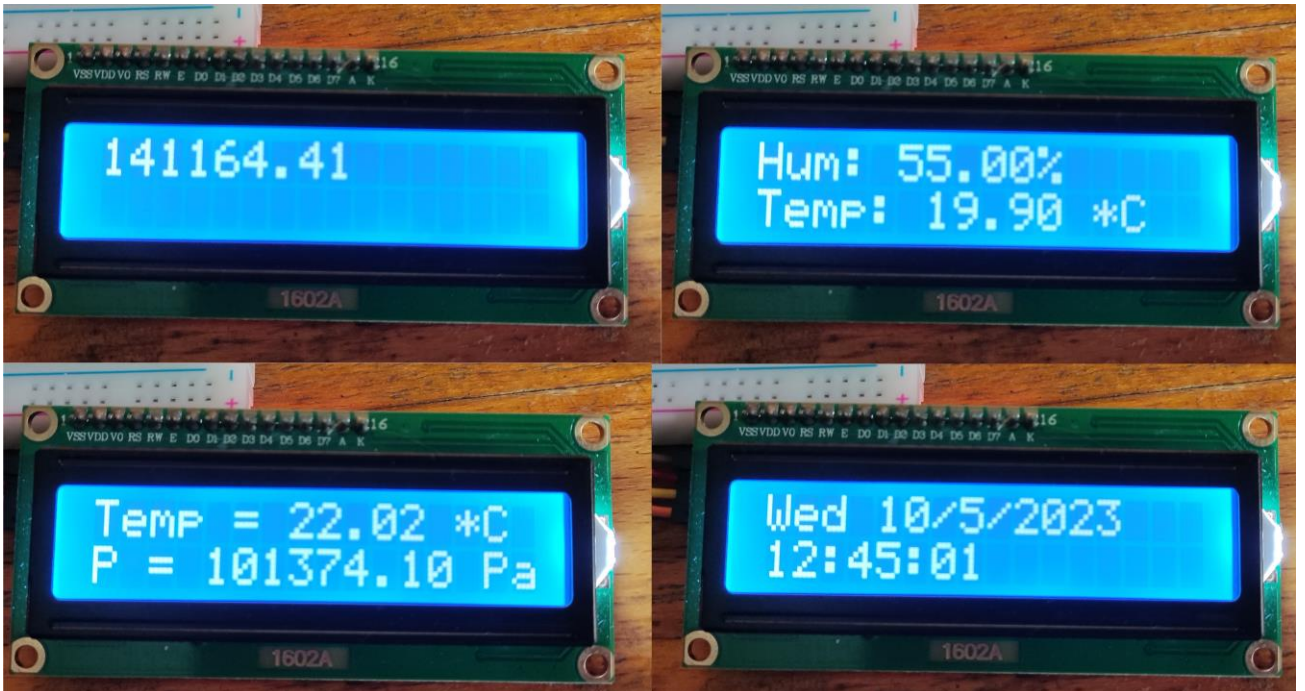


Рисунок 3.2 – Вивід інформації з датчиків на екран

## 3.2 Зміна режимів

### 3.2.1 Зміна режимів

Хоча циклічна зміна режимів дозволяє швидко отримати усі дані, користувачу може знадобитися меню для відображення даних з конкретного модулю. Для цього необхідно зробити так, щоб користувач міг змінювати модуль, за яким спостерігає ESP.

Усього буде закладено п'ять режимів: чотири на кожен з модулів та циклічний режим, це дозволить або спостерігати за певними показниками, або, як в першій версії, швидко отримати усі дані.

Зміна режимів буде відбуватися після натискання кнопки.

Щоб не конструювати громіздку схему, з кнопок, дротів та резисторів, на макетній платі, було вирішено використати матричну клавіатуру розміром 4x4. Щоб підключити клавіатуру до ESP необхідно з'єднати виходи клавіатури з цифровими виходами мікроконтролера, від

D0-го (перший вихід клавіатури) до D7-го (останній вихід клавіатури), ВКЛЮЧНО.

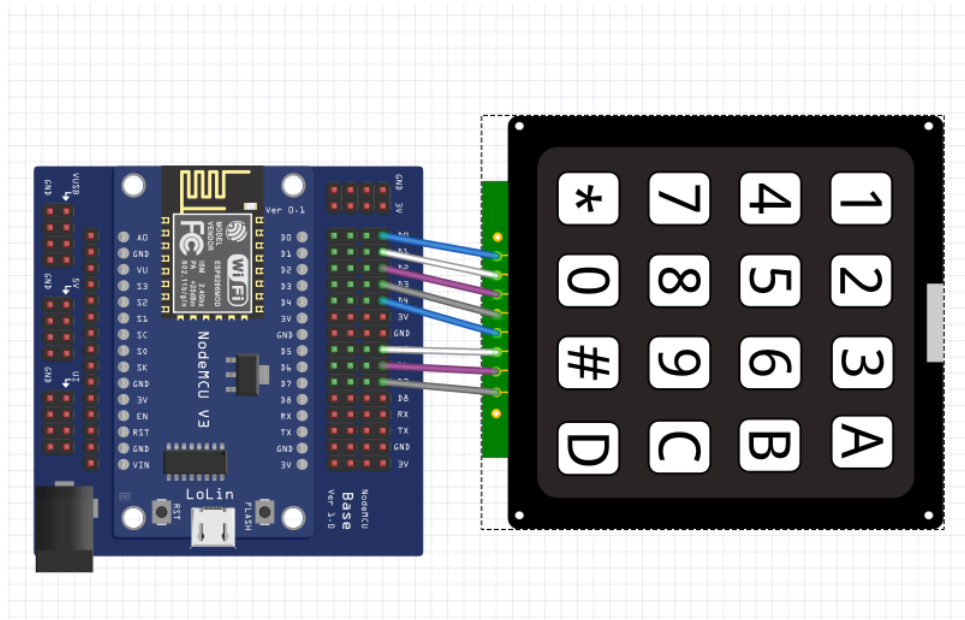


Рисунок 3.3 – Схема підключення матричної клавіатури до ESP

Зміна режимів відбувається таким чином: при натисканні на кнопку, змінна count збільшується на один і, в залежності від значення, обирається певний режим.

Перемикання режимів відбувається по колу, тобто після 5-го режиму відображення пристрій повертається до 1-го.

Лістинг 3.3 – Обробка натискання на кнопку:

```
void KeyBoard() {
    char myKey = myKeypad.getKey();

    if (myKey) {
        switch (myKey) {
            case 'A':
```

```
        count++;  
        break;  
    }  
}  
}
```

Щоб оптимізувати код, робота з усіма датчиками закладена у окремі функції, це дозволить змінювати окремі блоки без впливу на інші.

Функція зміни режимів реалізована через switch-case, який викликає функції для роботи з датчиками.

### 3.3 Функція актуалізації часу

Хоча пристрій обладнаний модулем реального часу, це не зберігає його від 2-х проблем:

1. Модуль DS3231 має погрішність  $\pm 2$  хвилини за рік;
2. Щоб зберігати поточний час, коли пристрій вимкнено, DS3231 використовує батарейку CR2032, доки вона не розрядиться. Далі час зупиняється на останньому значенні, до заміни батарейки.

Щоб усунути вказані проблеми, необхідно додати можливість актуалізації часу. Для цього ESP буде підключатися до NTP-серверу та отримувати дані про поточні дату та час, з точністю до секунди. Отримані дані будуть передаватися через функцію `adjust` до пам'яті DS3231. Після отримання інформації модуль продовжить відраховувати час від отриманої точки відліку.

Даний процес буде відбуватися лише при натисканні відповідної клавіші, тобто у той самий момент коли була натиснута клавіша.

Для підключення до NTP-серверу необхідне підключення до Інтернету, тому при натисканні клавіші ESP підключається до мережі Wi-Fi, до моменту передачі даних, після чого відключається від Wi-Fi.

### 3.4 Створення локального Web серверу

Останнім кроком, для завершення роботи над годинником-метеостанцією, стало створення локального серверу. На сервері будуть відображатися дані з усіх датчиків.

Перед створенням серверу необхідно змінити спосіб отримання даних з датчиків. Раніше дані отримувалися лише з конкретно обраного датчика, тобто якщо необхідно було побачити, наприклад, тиск, обирався відповідний режим. У рамках кожного режиму відбувалися заміри показників та виводилися на екран.

У даному випадку, щоб сервер правильно працював, усі заміри виконуються постійно, через кожні п'ять секунд, після чого надсилаються на сервер та виводяться на екран відповідно до обраного режиму. Такий спосіб дозволяє комбінувати вивід інформації на екран та надсилання даних на сервер без затримок та похибок у показниках.

Нижче наведено код функції, що отримує дані з датчиків.

Лістинг 3.4 – Код функції Stats:

```
if (millis() - timing > 5000) {  
    timing = millis();  
    h = dht.readHumidity();  
    t = dht.readTemperature();  
    PPM = gasSensor.getPPM();  
    pres = bmp.readPressure()*0.00750062;  
    if (!isnan(h) || !isnan(t)) {  
        prev_h = h;  
        prev_t = t;  
    }  
}
```

Для створення серверу знадобиться бібліотека “ESP8266WebServer”.

Щоб сервер запрацював необхідно налаштувати Access Point для ESP, тобто вказати назву та пароль мережі, яку буде «роздавати» ESP, налаштувати IP ESP, до якого будемо підключатися, мережевий шлюз та маску підмережі.

Далі залишиться прописати що буде коли на сервер зайшли та коли з'єднатися з сервером не вдалося.

При вдалому підключенні до серверу, будуть виводитися дані з датчиків. Кожні п'ять секунд буде оновлюватися сторінка. В інакшому випадку буде виводитися помилка 404 або інакшими словами на екрані з'явиться напис «Not found».

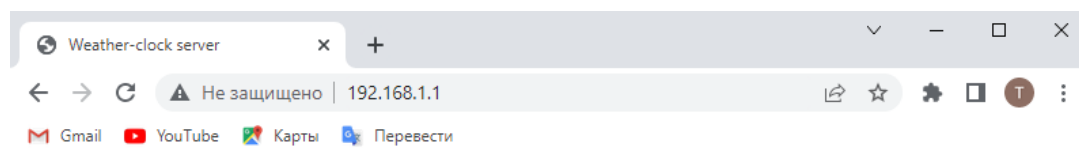


Рисунок 3.4 – Сторінка локального Web-серверу

Варто зауважити, хоча сервер має можливість вивести помилку 404, побачити помилку не вдасться через підключення до IP-адреси. У разі відсутності підключення буде виводитися стандартна браузерна сторінка. У рамках даного проекту не має потреби створювати сервер з підключенням за посиланням, достатньо, як вказано вище, IP-адреси.



Відправка даних на сервер відбувається через команду `server.send()`, яка отримує усі дані про відображення даних на сторінці зі `String` функції.

### 3.5 Тестування пристрою

Після завершення розробки необхідно протестувати пристрій. Пристрій тестувався як під час розробки, для оперативного усунення наявних помилок, так і після її завершення.

Тестування показало, що пристрій працює справно, але є декілька нюансів:

1. В усіх прикладах, до подібних проєктів, з використанням MQ135, налаштування датчику відбувається на основі середнього показника PPM у світі, тому на вулиці цей показник може бути нижчим, а у приміщенні схожий до середнього. Проте середній показник PPM у приміщенні дорівнює 800-1500 одиницям, в залежності від часу проведеного у цьому приміщенні. Щоб перевірити точність показників датчику необхідне відповідне обладнання. Під час роботи на данім проєктом, можливість придбати необхідне обладнання була відсутня;
2. Як було сказано у другому розділі, у пункті про матричну клавіатуру, клавіатура має неточне спрацьовування. Тобто при фізичному натисканні кнопка може спрацювати, може спрацювати інша кнопка, або реакції на натискання не буде узагалі. Причинами даної проблеми можуть бути:
  - неправильна обробка натискань у бібліотеці;
  - пошкодження конкретно обраної клавіатури при транспортуванні;
  - неякісна збірка клавіатури з заводу.

На момент написання пункту 3.5, а саме 28 травня 2023 року, точної причини виникнення проблем з клавіатурою виявлено не було.

3. Кожен раз коли подається напруга пристрій перевіряє 2 модулі. З малою імовірністю при перевірці ESP8266 може втратити адресу датчику BMP280, проблему можна вирішити повторною подачею напруги. Причини подібної поведінки станом на 28 травня 2023 року також невідомі.

Інших проблем у використанні представленого рішення помічено не було.

На рисунку нижче продемонстровано роботу пристрою, на екрані зображені поточні дата та час.

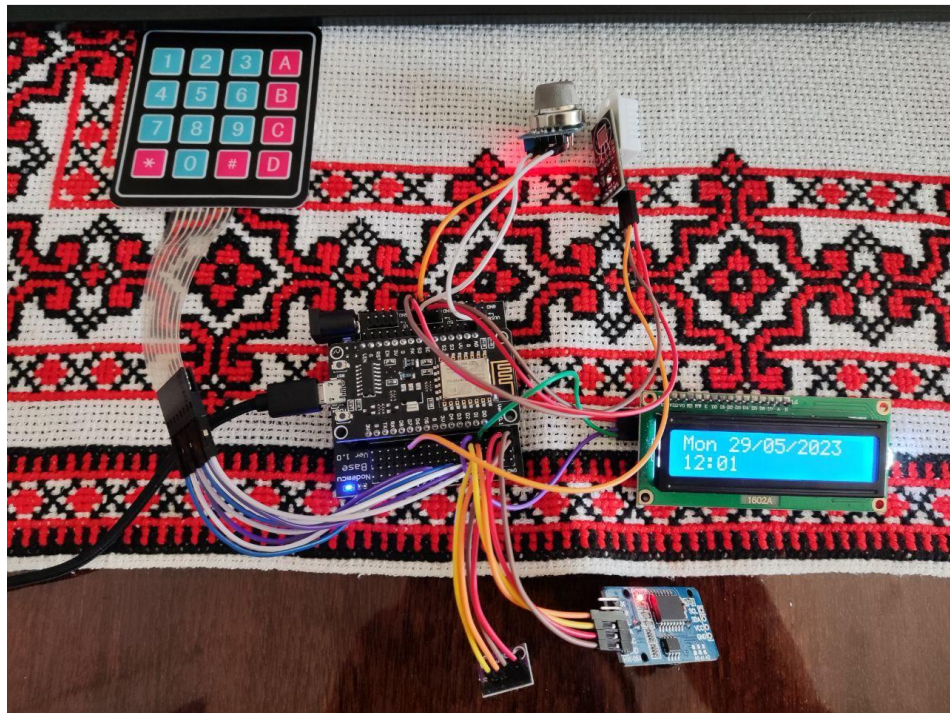
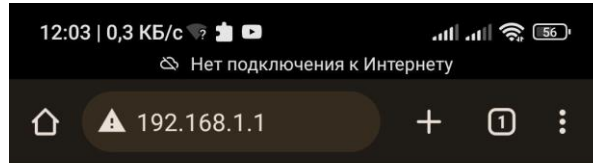


Рисунок 3.5 – Демонстрація роботи пристрою

Наразі функціонал пристрою не можна назвати об'ємним з декількох причин:

1. Матрична клавіатура, що обрана для даного проекту, задіяна лише на 12,5 відсотків. Залишилося 14 невикористаних кнопок, за допомогою яких можна значно розширити наявний функціонал;

2. Можливості серверу набагато більші ніж просте відображення інформації, а використаний спосіб створення серверу можна змінити на більш ефективний.



### Sensors data:

Temperature: 20°C

Humidity: 64%

Pressure: 737 mm Hg

CO2 level: 850ppm

Рисунок 3.6 – Дані з локального серверу

### 3.6 Висновки за розділом

Під час роботи над третім розділом було створено програмно-апаратний комплекс годинника-метеостанції. В ході розробки було замінено один датчик на більш точний та простіший у підключенні та додано додатковий модуль «Матрична клавіатура». Вказані зміни в апаратній складовій допомогли зменшити громіздкість конструкції та вилучити макетну плату з проекту.

Спеціально для даного пристрою було написано файл прошивки з наступним функціоналом:

- виконання замірів даних з кожного датчику та RTC модулю;
- режим для актуалізації часу;

- виведення отриманої інформації на екран;
- циклічна зміна режимів виводу інформації;
- веб-сервер на який надсилаються отримані дані.

Можливості даного пристрою більші ніж те що продемонстровано у рамках даної випускної роботи. Представлене рішення може повноцінно використовуватись, проте необхідно розширити наявний функціонал та виправити проблеми вказані раніше.

## ВИСНОВКИ

За період виконання випускної роботи було переглянуто різні варіанти виконання метеостанцій, проаналізовано можливості таких пристроїв та створено власне рішення годинника-метеостанції.

Щоб виконати поставлену задачу, було використано такі апаратні компоненти:

- Плата мікроконтролеру ESP8266;
- Датчик вологості та температури DHT22;
- Датчик газу MQ-135;
- RTC-модуль DS3231;
- Датчик тиску BMP280;
- Рідкокристалічний дисплей LCD1602;
- Плата розширення для ESP8266;
- Матрична клавіатура 4x4.

У створеному пристрої закладено такі можливості:

Програмний код був створений у середовищі програмування Arduino IDE версії 1.8.19.

Представлений пристрій має модульну систему, що дозволяє розширювати функціонал, змінюючи програмний код та додаючи нові датчики, або змінюючи поточну конфігурацію на більш точні.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Електронні проекти на основі ESP8266 та ESP32 / [Електронний ресурс] / Режим доступу www. URL: <https://coollib.com/b/585724-nil-kameron-elektronnyie-proektyi-na-osnove-esp8266-i-esp32-sozdanie-prilozheniy-i-ustroystv-s-podde.pdf>
2. Інтернет речей з ESP8266 / [Електронний ресурс] / Режим доступу www. URL: <https://djvu.online/file/y3swet5odkmxs>
3. Метеорологічна станція / [Електронний ресурс] / Режим доступу www. URL: [https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B5%D0%BE%D1%80%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B0\\_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D1%8F](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B5%D0%BE%D1%80%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%87%D0%BD%D0%B0_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D1%8F)
4. Метеостанція як вибрати? / [Електронний ресурс] / Режим доступу www. URL: <https://simvolt.ua/meteostantsiya-yak-vibrati/>
5. Топ-9 кращих домашніх метеостанцій / [Електронний ресурс] / Режим доступу www. URL: <https://znai.com.ua/top-9-krashchih-domashnih-meteostancy/>
6. Датчик газу MQ-135 / [Електронний ресурс] / Режим доступу www. URL: <https://www.mini-tech.com.ua/datchik-gaza-mq-135-modul>
7. Барометр BMP280 / [Електронний ресурс] / Режим доступу www. URL: <https://arduino.ua/ru/prod1758-barometr-datchik-atmosfernogo-davleniya-na-bmp280>
8. DS3231 RTC Модуль годин реального часу / [Електронний ресурс] / Режим доступу www. URL: [https://3v3.com.ua/product\\_7518.html](https://3v3.com.ua/product_7518.html)
9. Характеристики и підключення датчика DHT11 / [Електронний ресурс] / Режим доступу www. URL: <https://www.arduinka.top/start/stat5>

10. Підключення LCD 1602 до Arduino / [Електронний ресурс] / Режим доступу www. URL: <https://hi-news.pp.ua/tehnka-tehnologyi/13951-pdklyuchennya-lcd-1602-do-arduino-opis-funkcyi-nstrukcyi-osoblivost-problemi-yih-rshennya.html>
11. NodeMCU ESP8266 Detailed Review / [Електронний ресурс] / Режим доступу www. URL: <https://www.make-it.ca/nodemcu-details-specifications/>
12. NodeMCU Base-board / [Електронний ресурс] / Режим доступу www. URL: <https://www.handsontec.com/dataspecs/module/NodeMCU%20Base-board.pdf>
13. ESP8266: Interfacing with a 4×4 Matrix Keypad / [Електронний ресурс] / Режим доступу www. URL: <https://techtutorialsx.com/2017/03/18/esp8266-interfacing-with-a-4x4-matrix-keypad/>

## ДОДАТКИ

## ДОДАТОК А. ВИХІДНИЙ КОД ПРОГРАМИ

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <MQ135.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <Adafruit_BMP280.h>
#include <Keypad.h>
#include "DHT.h"
#include "RTClib.h"

#define MODE_AM 4
#define DHTPIN 13
#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_MOSI 11
#define BMP_CS 10
#define DHTTYPE DHT22

IPAddress local_ip(192,168,1,1);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);
```



```
ESP8266WebServer server(80);

Adafruit_BMP280 bmp;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C LCD (0x27, 16, 2);
RTC_DS3231 rtc;
MQ135 gasSensor = MQ135(A0);

const char* ssid = "Poco"; // your network SSID (name)
const char* passwd = "82581488";

const char* local_ssid = "NodeMCU"; // SSID
const char* local_password = "12345678";

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri",
"Sat"};

const byte n_rows = 4;
const byte n_cols = 4;

char keys[n_rows][n_cols] = {
  {'1','4','7','*'},
  {'2','5','8','0'},
  {'3','6','9','#'},
  {'A','B','C','D'}
};

byte rowPins[n_rows] = {0, 4, 5, 16};
```

```
byte colPins[n_cols] = {13, 12, 14, 2};
```

```
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "pool.ntp.org");
```

```
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, n_rows,  
n_cols);
```

```
int count;
```

```
int State;
```

```
float h;
```

```
float t;
```

```
float prev_h;
```

```
float prev_t;
```

```
float PPM;
```

```
float pres;
```

```
unsigned long timing;
```

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
    Serial.begin(115200);
```

```
    LCD.init();
```

```
    LCD.backlight();
```

```
    LCD.setCursor(0, 0);
```

```
    LCD.print("BMP test!");
```

```
    delay(2000);
```

```
    if (!bmp.begin(0x76)) {
```

```
        LCD.setCursor(0, 0);
```

```
LCD.print("BMP not found!");  
LCD.setCursor(0, 1);  
LCD.print("Change SensorID!");  
while (1) delay(10);  
}
```

```
else{  
  LCD.setCursor(0, 1);  
  LCD.print("BMP found!");  
  delay(2000);  
}
```

```
LCD.clear();  
LCD.setCursor(0, 0);  
LCD.print("RTC test!");  
delay(2000);
```

```
if (! rtc.begin()) {  
  LCD.setCursor(0, 0);  
  LCD.print("RTC not found!");  
  LCD.setCursor(0, 1);  
  LCD.flush();  
  while (1) delay(10);  
}
```

```
else {  
  LCD.setCursor(0, 1);  
  LCD.print("RTC found!");
```

```
    delay(2000);
  }

  pinMode(DHTPIN, INPUT);
  dht.begin();

  WiFi.softAP(local_ssid, local_password);
  WiFi.softAPConfig(local_ip, gateway, subnet);
  delay(100);

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);

  server.begin();
  LCD.clear();
  LCD.setCursor(0, 0);
  LCD.print("Server started!");
  delay(2000);

  timeClient.begin();
  timeClient.setTimeOffset(10800);
}

void loop() {
  // put your main code here, to run repeatedly:
  KeyBoard();
  switch (State) {
    case 0:
```

```
Stats();
modes();
State = 1;
case 1:
if (millis() - timing > 5000) {
    timing = millis();
    Stats();
    modes();
}
}
server.handleClient();
}

void Stats() {
    h = dht.readHumidity();
    t = dht.readTemperature();
    PPM = gasSensor.getPPM();
    pres = bmp.readPressure()*0.00750062;
    if (!isnan(h) || !isnan(t)) {
        prev_h = h;
        prev_t = t;
    }
}

void Dht() {
    if (isnan(h) || isnan(t)) {
        if (prev_h == 0 || prev_t == 0) {
```

```
LCD.clear();
LCD.setCursor(0, 0);
LCD.print("DHT read failed!");
}
else {
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print("Hum: ");
    LCD.print(prev_h);
    LCD.print("%");
    LCD.setCursor(0,1);
    LCD.print("Temp: ");
    LCD.print(prev_t);
    LCD.print("*C ");
}
}

else {
LCD.clear();
LCD.setCursor(0,0);
LCD.print("Hum: ");
LCD.print(h);
LCD.print("%");
LCD.setCursor(0,1);
LCD.print("Temp: ");
LCD.print(t);
LCD.print("*C ");
}
}
```

```
}

void MQ() {
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print("CO2 level:");
    LCD.setCursor(0,1);
    LCD.print(PPM);
    LCD.print("ppm");
}

void BMP() {
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print("Pressure:");
    LCD.setCursor(0,1);
    LCD.print(int(pres));
    LCD.print("mm");
}

void RTC() {
    DateTime now = rtc.now();
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print(daysOfTheWeek[now.dayOfTheWeek()]);
    LCD.print(" ");
    if (now.day() < 10) LCD.print("0");
    LCD.print(now.day(), DEC);
}
```

```
LCD.print('/');
if (now.month() < 10) LCD.print("0");
LCD.print(now.month(), DEC);
LCD.print('/');
LCD.print(now.year(), DEC);

LCD.setCursor(0,1);
if (now.hour() < 10) LCD.print("0");
    LCD.print(now.hour(), DEC);
    LCD.print(':');
if (now.minute() < 10) LCD.print("0");
    LCD.print(now.minute(), DEC);
}

void curTime() {
    int tries = 15;
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Connect to WiFi");

    WiFi.begin(ssid, passwd);
    LCD.setCursor(0, 1);
    while (--tries && WiFi.status() != WL_CONNECTED) {
        delay(500);
        LCD.print(".");
    }

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



```
LCD.clear();  
LCD.setCursor(0, 0);  
LCD.print("Failed");  
delay(1000);  
}
```

```
if (WiFi.status() == WL_CONNECTED) {  
LCD.clear();  
LCD.setCursor(0, 0);  
LCD.print("Success");  
delay(1000);
```

```
timeClient.update();
```

```
time_t epochTime = timeClient.getEpochTime();  
int curHour = timeClient.getHours();  
int curMin = timeClient.getMinutes();  
int curSec = timeClient.getSeconds();  
struct tm *ptm = gmtime ((time_t *)&epochTime);  
int monthDay = ptm->tm_mday;  
int curMon = ptm->tm_mon+1;  
int curYear = ptm->tm_year+1900;
```

```
rtc.adjust(DateTime(curYear, curMon, monthDay, curHour, curMin, curSec));
```

```
WiFi.disconnect();  
}
```

```
else {
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Failed");
    delay(1000);
}
}

void KeyBoard() {
    char myKey = myKeypad.getKey();
    delay(100);
    if (myKey) {
        switch (myKey) {

            case 'B':
                curTime();
                break;

            case 'C':
                State = 0;
                if (++count >= MODE_AM) count = 0;
                break;
        }
    }
}

void modes() {
    switch (count) {
```

```
    case 0:
    RTC();
    break;

    case 1:
    BMP();
    break;

    case 2:
    MQ();
    break;

    case 3:
    Dht();
    break;
}
}

void handle_OnConnect() {
    server.send(200, "text/html", SendHTML(t, h, prev_h, prev_t, pres, PPM));
}

void handle_NotFound(){
    server.send(404, "text/plain", "Not found");
}

String SendHTML(float Temperaturestat,float Humiditystat, float PrevTemp,
float PrevHum, float PressureStat, float PPMstat){
```

```

String ptr = "<!DOCTYPE html> <html>\n";
ptr += "<meta http-equiv=\"refresh\" content=\"5\" ><head><meta
name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-
scalable=no\">\n";
ptr += "<title>Weather-clock server</title>\n";
ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center; }\n";
ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto
30px;}\n";
ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";
ptr += "</style>\n";
ptr += "</head>\n";
ptr += "<body>\n";
ptr += "<div id=\"webpage\">\n";
ptr += "<h1>Sensors data:</h1>\n";
if (isnan(Temperaturestat) || isnan(Humiditystat)) {
    ptr += "<p>Temperature: ";
    ptr += (int)PrevTemp;
    ptr += "&deg;C</p>";
    ptr += "<p>Humidity: ";
    ptr += (int)PrevHum;
    ptr += "%</p>";
}
else {
    ptr += "<p>Temperature: ";
    ptr += (int)Temperaturestat;
    ptr += "&deg;C</p>";
    ptr += "<p>Humidity: ";

```

```
    ptr +=(int)Humiditystat;
    ptr += "%</p>";
}
ptr += "<p>Pressure: ";
ptr +=(int)PressureStat;
ptr += " mm Hg</p>";
ptr += "<p>CO2 level: ";
ptr +=(int)PPMstat;
ptr += "ppm</p>";
ptr += "</div>\n";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}
```