

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету)

Кафедра автоматики та телекомунікацій
(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри автоматики та
телекомунікацій

_____ (підпис)

_____ (ім'я та прізвище)

«__» _____ 2023 р.

Випускна кваліфікаційна робота

магістра

(освітньо-кваліфікаційний рівень)

на тему «Дослідження та розробка системи автоматичного керування метеостанцією»

Виконав студент 2 курсу, групи СУАм-22

(шифр групи)

спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

Рогожин Андрій Володимирович

(Прізвище, ім'я та По-батькові)

(підпис)

Керівник доцент кафедри АТ, к.т.н., _____ Ганна ТЕЛИЧКО

(посада, науковий ступінь, вчене звання, прізвище та ім'я)

(підпис)

Рецензент доцент кафедри ЕТ, к.т.н., доцент _____ Олександр ШТЕПА

(посада, науковий ступінь, вчене звання, прізвище та ім'я)

(підпис)

Рецензент зав.каф ЕЛПН, к.т.н., доцент _____ Олександр КОЛЛАРОВ

(посада, науковий ступінь, вчене звання, прізвище та ім'я)

(підпис)

Нормоконтроль:

Дар'я ЖУКОВСЬКА

20.12.2023 р.

Засвідчую, що у цій випускній кваліфікаційній
роботі немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

(підпис)

Луцьк – 2023 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету)

Кафедра автоматики та телекомунікацій
(повна назва кафедри)

Захист відбувся _____
(дата)

з оцінкою _____

Секретар ЕК _____
(підпис)

Випускна кваліфікаційна робота
бакалавра

Тема: «Дослідження та розробка системи автоматичного керування метеостанцією»

Спецчастина: _____

Виконавець, студент
гр. СУАм-22

_____ Андрій РОГОЖИН
(підпис, дата, Ім'я ПРІЗВИЩЕ)

Керівник

_____ Ганна ТЕЛИЧКО
(підпис, дата, Ім'я ПРІЗВИЩЕ)

Консультанти:

_____ Ганна ТЕЛИЧКО
(підпис, дата, Ім'я ПРІЗВИЩЕ)

_____ Валерій ПОЦЕПАСВ
(підпис, дата, Ім'я ПРІЗВИЩЕ)

_____ Дар'я ЖУКОВСЬКА
(підпис, дата, Ім'я ПРІЗВИЩЕ)

Нормоконтроль

_____ Дар'я ЖУКОВСЬКА
(підпис, дата, Ім'я ПРІЗВИЩЕ)

Луцьк – 2023 р.

**Державний вищий навчальний заклад
"Донецький національний технічний університет"**

Інститут, факультет Комп'ютерно-інформаційних технологій та автоматизації
Кафедра Автоматика та телекомунікації
Освітньо-кваліфікаційний рівень магістр
Галузі знань 15 Автоматизація та приладобудування
(шифр і назва)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри АТ

« _____ » 2023 року

З А В Д А Н Н Я
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Рогожин Андрій Володимирович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Дослідження та розробка системи автоматичного керування метеостанцією

керівник проекту (роботи) Теличко Ганна Олександрівна, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 22.09.2023 року № 478

2. Строк подання студентом проекту (роботи) _____

3. Вихідні дані до проекту (роботи) _____
Технічна документація та матеріали з переддипломної практики

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд і аналіз існуючих автоматичних систем метеостанцій, розробка функціональної схеми, вибір елементної бази, розробка програмного забезпечення, розробка конструкції.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Функціональна схема, блок-схема BMP280, схема підключення датчика
температури, вологості та тиску BME280 до WeMos D1 mini

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Теличко Г. О., к.т.н.,доц.		
2	Жуковська Д.О., асистент		
3	Поцепаєв В.В., зав.каф., к.т.н.,доц.		
Нормо-контроль	Жуковська Д.О., асистент	-	20.12.2023 р.

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналітичний огляд	01.10.22 – 13.10.22	
2	Обґрунтування та вибір компонентів	15.10.22 – 29.10.22	
3	Проектування приладу і написання програмного коду для wemos d1 mini	29.10.22 – 13.11.22	
4	Охорона праці та безпека під час надзвичайних ситуацій на підприємстві	15.11.22 – 09.12.22	
5	Представлення дипломної роботи науковому керівнику	16.12.22	
6	Захист дипломної роботи	27.12.22	

Студент

(підпис)

Андрій Рогожин

(прізвище та ініціали)

Керівник роботи

(підпис)

Ганна Теличко

(прізвище та ініціали)

ЛИСТ ЗАУВАЖЕНЬ

Посада, Ім'я та ПРІЗВИЩЕ	Суть зауваження, оцінка та підпис

АНОТАЦІЯ

Рогожин А.В. Дослідження та розробка системи автоматичного керування метеостанцією / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» зі спеціальності 151 автоматизація та комп'ютерно-інтегровані технології. – ДВНЗ «ДонНТУ», Луцьк, 2023.

Пояснювальна записка: 115 стор., 54 рис., 2 табл., 45 посилання.

У процесі написання дипломної роботи було проаналізовано роботу автоматизованої метеостанції, керування яких виконується за допомоги мікроконтролера. Для цього були розглянуті сучасні станції для моніторингу погоди. У процесі роботи було проведено огляд датчиків і вибір мікроконтролерів за допомогою якого буде здійснюватися управління макетом.

Обґрунтований вибір компонентів включав детальний аналіз кожного елемента для складової бази. Крім того, був розроблений план монтажу зовнішнього корпусу, а також були розроблені та детально описані електричні та функціональні схеми. Особлива увага була приділена підключенню сенсорів до макетної плати; інформація про це представлена на схемах. Наша автоматизована метеостанція вимірює температуру, тиск і вологість повітря. Відзначимо, що ця станція ідеально підходить для використання в невеликих приміщеннях, де не потрібно робити вимірювання з високою точністю.

Ключові слова: АВТОМАТИЗОВАНА МЕТЕОСТАНЦІЯ, МІКРОКОНТРОЛЕР, ДАТЧИК ВИМІРЮВАННЯ, ESP8266, BME280.

Перелік публікацій: Розробка системи автоматичного керування водопостачанням віддаленого селища / Г.О. Теличко, Д.О. Романенков, Б.Є. Карєв, А.В. Рогожин // Науковий вісник ДонНТУ/ – 2022. –№ 1(8)-2(9). – С. 119-127

Розробка метіостанції з передачею даних на веббраузер / Рогожин А.В.,Теличко Г.О// Всеукраїнський науково-практичний форум «ТАК2023»: телекомунікації, автоматизація, комп'ютерно-інформаційні технології, 5-6 грудня 2023 р.: збірка доповідей. – ДВНЗ «ДонНТУ», м. Луцьк, 2023. – С. 180 – 182.

ABSTRACT

Rogozhyn A.V. Research and development of a system of automatic control of a weather station / Graduation qualification work for obtaining an educational degree "Master" specialty 151 «Automation and computer-integrated technologies" - SHEE DonNTU Lutsk, 2023.

Explanatory note: 115 pages, 54 figures, 2 tables, 45 references.

In the process of writing the thesis, the work of an automated weather station, which is controlled by a microcontroller, was analyzed. For this, modern weather monitoring stations were considered. In the process of work, an overview of sensors and selection of microcontrollers was carried out in order to establish control of the layout.

Reasoned selection of components included a detailed analysis of each element for the component base. In addition, an external enclosure assembly plan was developed, and electrical and functional schematics were developed and detailed.

Special attention was paid to connecting the sensors to the breadboard; information about this is presented in the diagrams. Our automated weather station measures temperature, pressure and air humidity. Note that this station is ideal for use in small rooms where high-precision measurements are not required.

Keywords: AUTOMATED WEATHER STATION, MICROCONTROLLER, MEASUREMENT SENSOR, ESP8266, BME280.

Publisher publication list: Development of a system of automatic management of water supply of a remote village / G.O. Telychko, D.O. Romanenkov, B.E. Karev, A.V. Rogozhyn // Scientific Bulletin of DonNTU/ – 2022. – No. 1(8)-2(9). - P. 119-127

Development of a weather station with data transfer to a web browser / A.V. Rogozhyn, G.O. Telychko// All-Ukrainian scientific and practical forum "TAK2023": telecommunications, automation, computer and information technologies, December 5-6, 2023: a collection of reports . - DonNTU Higher Secondary School, Lutsk, 2023. - pp. 180-182.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП.....	10
1 АНАЛІТИЧНИЙ ОГЛЯД	12
1.1 Розвиток метеорологічних пристроїв.....	12
1.2 Приклади сучасних метеостанції	31
Висновки до розділу 1	43
2 ОБҐРУНТУВАННЯ ТА ВИБІР КОМПОНЕНТІВ.....	44
2.1 Функціональна схема приладу	44
2.2 Вимоги для елементної бази приладу	45
2.3 Вибір мови програмування для мікроконтролера WeMos D1 mini ...	61
Висновки до розділу 2	70
3 ПРОЕКТУВАННЯ ПРИЛАДУ І НАПИСАННЯ ПРОГРАМНОГО КОДУ	
WEMOS D1 MINI.....	71
3.1 Підключення датчика BME280 до плати WeMos D1 mini	71
3.2 Підготовка IDE Arduino	72
3.3 Установка бібліотеки для сенсора BME280	74
3.4 Опис макету.....	76
3.5 Створення програмного продукту.....	77
Висновки до розділу 3	90
ВИСНОВОК	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	94
ДОДАТОК А. Охорона праці та безпека в надзвичайних ситуаціях	99
ДОДАТОК Б. Перша версія коду	109
ДОДАТОК В. Друга версія коду	112

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БАММ - барометр-анероїд мембранний метеорологічний.

АМ - автоматизована метеостанція.

АДМ – автоматична дорожня метеостанція.

ПЗ – програмне забезпечення.

ЛСМ -лісова метеорологічна станція.

АМ - автоматизована метеостанція.

ARM - advanced RISC machine.

IDE - integrated drive electronics.

ГМС - гідрологічна метеорологічна станція.

USB - universal serial bus.

АДМ – автоматична дорожня метеостанція.

ПК – персональний комп'ютер.

RISC - reduced instruction set computing.

COM – communication port, послідовний порт.

ВСТУП

Клімат і погода можуть змінюватися протягом короткого проміжку часу, який може становити години або дні. У минулому традиційні знання корінних народів дозволяли передбачати погоду. Це могло допомогти передбачити, коли очікувати дощі в певному місці протягом року. З іншого боку, такі методи прогнозування в наш час стали неактуальні у зв'язку з їхньою ефективністю.

Погода, яку спостерігають протягом тривалого часу, надає інформацію про клімат регіону. Всесвітня метеорологічна організація рекомендує визначити клімат як середній стан погодних параметрів протягом тридцяти років. Протягом останніх сто років глобальне потепління та збільшення кількості опадів призвели до повеней і посухи по всьому світу. Це свідчить про зміни клімату та викликає занепокоєння в усьому світі. Сезонні дощі приходять раніше або пізніше в деяких частинах Східної Африки, часто вони менше за те що очікували і дуже погано розподіляються. Через посуху в бідних країнах виникли проблеми з голодом. Згідно з дослідженням економічного впливу посухи на сільське господарство, валовий внутрішній продукт (ВВП) знизився на 5%, а промисловість зазнала втрат.

Метеорологія — це наука, яка вивчає погоду, клімат тощо. Національна економіка, особливо сільська, залежить від метеорологічних явищ. У нинішньому світі, де традиційні знання не завжди корисні та погода може змінитися в будь-який час, важливо мати надійні та точні дані. Для збору точних даних і кращої інтерпретації цих даних є необхідність розширення мереж метеостанцій. Однак у менш розвинених країнах це може бути неможливим через недостатнього фінансування проектів зв'язані з метеостанціями і збору даних традиційним методом.

В даний час дуже популярні портативні пристрої, що базуються на мікроконтролерах і оснащені багатьма датчиками. У минулому використовували традиційні методи прогнозування погоди. В даний час використовують

автоматизовані метеостанції (АМ), які обладнані датчиками, які автоматично збирають і передають погодні дані. В цій роботі була поставлена задача розробити економічну метеостанцію, яка базується на міконтролері та обладнана датчиками для вимірювань.

У цьому проекті WeMos D1 mini, який має вбудовану мікросхему ESP8266, виконує функцію пристрою управління, підключається до існуючої мережі Wi-Fi і запускає процес створення веб-сервера. Коли будь-який з'єднаний пристрій звертається до цього веб-сервера, ESP8266 отримує дані про температуру, вологість, атмосферний тиск і висоту від BME280. Після цього він відображає ці вимірювання у веб-браузері пристрою.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Розвиток метеорологічних пристроїв

Метеорологічні пристрої були розроблені ще в XVII–XIX столітті. Наразі метеорологічне приладдя демонструє значний прогрес. Кожен день розробляються прилади, які використовують сучасні технології, радіозв'язок і радіолокацію, для виявлення метеорологічних змін. Автоматизовані станції для спостереження та передачі результатів досліджень також активно впроваджується, оскільки вони не потребують прямого втручання людини.

Метеорологічні прилади можуть виконувати негайні вимірювання та постійне реєстрування зазначених параметрів у часі, зазвичай у вигляді графіків або кривих. Перші включають прилади для вимірювання температури та тиску, а останні включають термографи та барографи [1].

1.1.1 Розвиток термометрів

У 1641 році Фернандо Медичі, природознавець і винахідник, який вивчав науку під керівництвом Галілея, розробив перший герметичний термометр для великого герцога Тоскани. Невелика скляна куля була прикріплена до скляної трубки, яка представляла кінцевий вигляд термоскопа Галілея, як показано на рис. 1.1 [2].

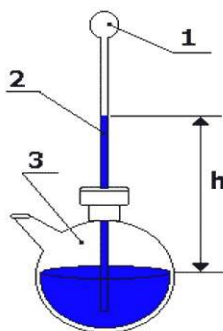


Рисунок 1.1 – Термоскоп Галілея

Щоб провести вимірювання кульку 1 нагрівали; це в свою чергу приводило до розширювання повітря в кульці. Згодом розширене повітря виходило в атмосферу через трубку 2. Як наслідок цього було створено конструкцію, представлену на малюнку 1.1. В результаті повітря в кульці поступово остигало до температури повітря де находилась куля, а це в свою чергу приводило до стиснення. Воду з посудини 3 піднімали по трубці 2 на певну висоту через атмосферний тиск.

Медичі створив пристрій з високою чутливістю, використовуючи кулю, з'єднану з тонкою трубкою (зображено на рис. 1.2), використовуючи основний принцип пристрою Галілея [3]. Основна відмінність винаходу Медичі полягала в тому, що куля наповнювалася термометричною рідиною, а не повітрям. Для вимірювання зміни об'єму цієї рідини під час нагрівання використовували рівномірну шкалу, нанесену на трубку, подібна конструкція шкали використовувались у термометрах Санторіо і Сагредо.

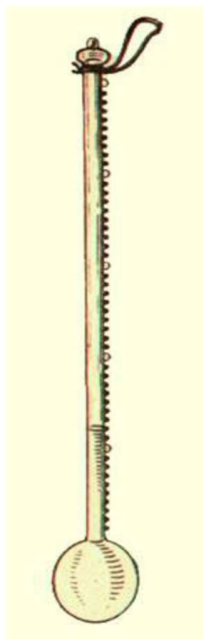


Рисунок 1.2 – Термометр Фернандо Медичі

Рідинний термометр може працювати в межах температурного інтервалу, обмеженого атмосферним тиском і точкою замерзання рідини. Медича вибрав винний спирт як термометричну рідину, оскільки основною метою було вимірювання температури атмосферного повітря. Точка кипіння винного спирту (78 °C) була ідеальною для цілей Медичі, а точка замерзання (-114 °C) вважалася недосяжною, що робило її практично незамерзаючою рідиною для цієї метеорологічної системи.

У термометрах, які виготовляли в майстерні Медичі, шкалою були або намистинки, припаяні до трубки, або точки, нанесені розплавленою емаллю на розігріту трубку. Зазвичай шкала мала п'ятдесят поділок, де десять відповідало тайнню снігу, а сорок — прирівнювалось до максимального нагріву приладу на сонці [3].

У 1724 році Габріель Фаренгейт, німецький виробник приладів, визначив, що ртуть є найкращою рідиною для вимірювання температури. Використовуючи суміш льоду та солоної води як точку відліку, він калібрував свій перший термометр таким чином. Температура замерзання звичайної води набагато нижча, ніж у солоної води. Отже, Фаренгейт визначив температуру, при якій солена вода замерзає, за розрахунками яка складала 32 градуси, а температура тіла здорової людини – 96 градусів. У результаті використання цих точок він досяг температури кипіння води на рівні моря в 212 градусів. Після цього він підвищив температуру замерзання води до 32 градусів. Таким чином, він розширив діапазон температур між кипінням води та замерзанням на 180 градусів.

Однак цифра 180 залишалася незручною для вимірювання температури. Отже, двадцять років пізніше шведський природознавець Карл Лінней, який був відомий своєю системою таксономії, він використовував її для іменування видів, і шведський астроном Андерс Цельсій окремо розробили шкалу, яка охоплювала весь діапазон температур в 100 градусів між точками замерзання та кипіння води.

Ця шкала була названа «шкалою Цельсія» через те, що між цими двома точками було сто поділок.

Трохи більше ста років по тому, в 1848 році, лорд Кельвін почав досліджувати теорію тепла та значно ширший діапазон температур. Хоча він використовував шкалу за Цельсієм, він почав з абсолютного нуля — найнижчої можливої температури у Всесвіті — де зупиняється рух кожної молекули. Це становило $-273,16\text{ }^{\circ}\text{C}$. Шкалу Ренкіна можна використовувати для визначення абсолютної температури за Фаренгейтом. У 1948 році Міжнародна конференція прийняла шкалу Цельсія як стандартну міру, але старі методи залишилися актуальними [4]. У Великобританії та Сполучених Штатах Фаренгейт все ще використовується.

Температуру повітря зазвичай вимірюють на висоті приблизно 2 метри над поверхнею землі. Тим не менш, є винятки для аналізу стану приземного шару повітря. У таких ситуаціях термометри можуть бути розташовані на різних висотах. На суднах можна змінювати розміщення термометрів, щоб вимірювати температуру в різних місцях.

Термометр повинен бути захищений від прямого сонячного випромінювання, а також від випромінювання, яке випромінюється з поверхні Землі та навколишніх об'єктів, таких як будівлі та рослини, щоб зробити правильні вимірювання температури. Це досягається шляхом встановлення термометра в спеціальні сконструйовані корпуси. Завдяки цьому Термометр може зберігати температуру, яка відповідає температурі навколишнього повітря в цих умовах. Термометр який знаходиться без корпусу нагріється значно швидше, ніж той що в корпусі, тому що він не встигає поміряти температуру повітряного шар навколо нього, тому він не буде показувати точну температуру. Зазвичай використовують дерев'яний корпус який фарбують білою фарбою, щоб відбити сонячні промені та уникнути перегріву [5].

Передача температурних даних у вигляді електричних величин є найбільш цілеспрямованою для автоматизованих систем, таких як кліматичні. Ці величини можуть бути постійними або змінними (сигналами), що означає, що вони можуть бути або постійними, або змінними сигналами. З цієї причини існують цифрові та аналогові датчики температури. Обидва різновиди використовуються в кліматичних середовищах [6].

Термопари, інфрачервоні термометри, терморезистори, біметалеві термометри та інші пристрої можна використовувати для вимірювання температури. Біметалеві термометри складаються з двох тонких смужок різних металів, склеєних разом (рис. 1.3).

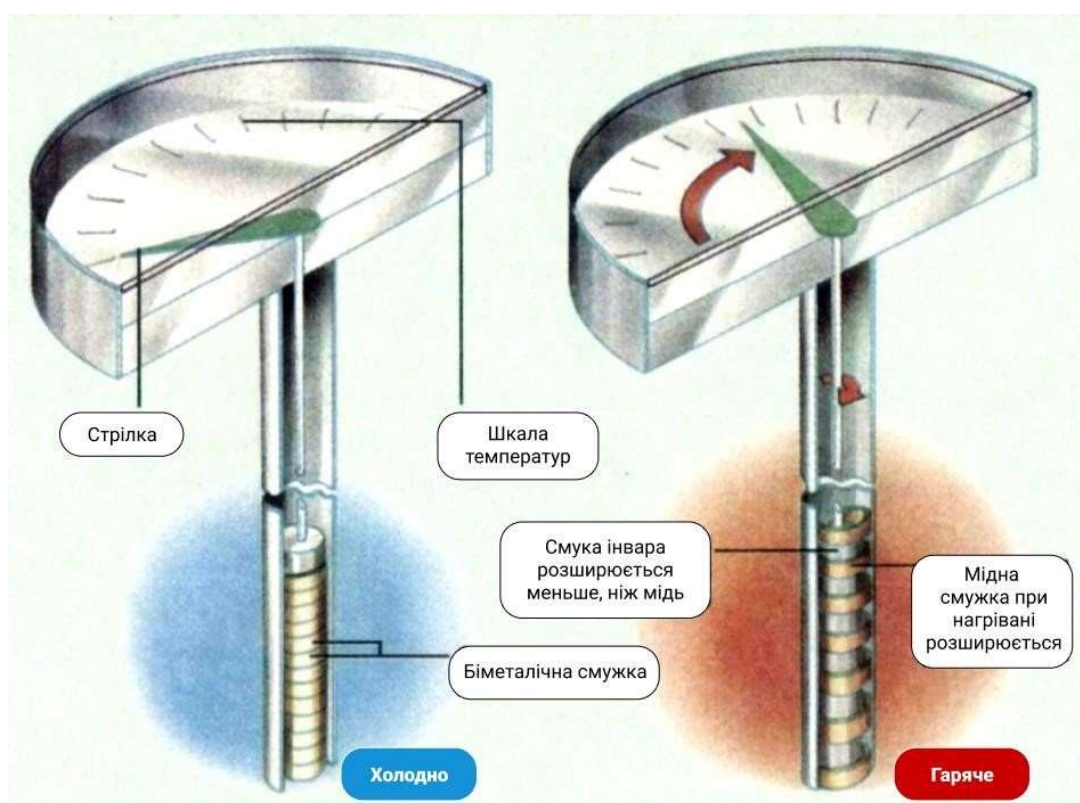


Рисунок 1.3 – Принцип роботи та вигляд біметалічного термометра

Біметалеві термометри мають дві смужки, які не рухаються одна відносно одної. Біметалеву смугу можна описати як пряму консольну балку. Ця балка має один фіксований кінець, але коли температура змінюється, її вільний кінець відхиляється. У метеорології все частіше використовують електронні датчики температури, як показано на рисунку 1.4.

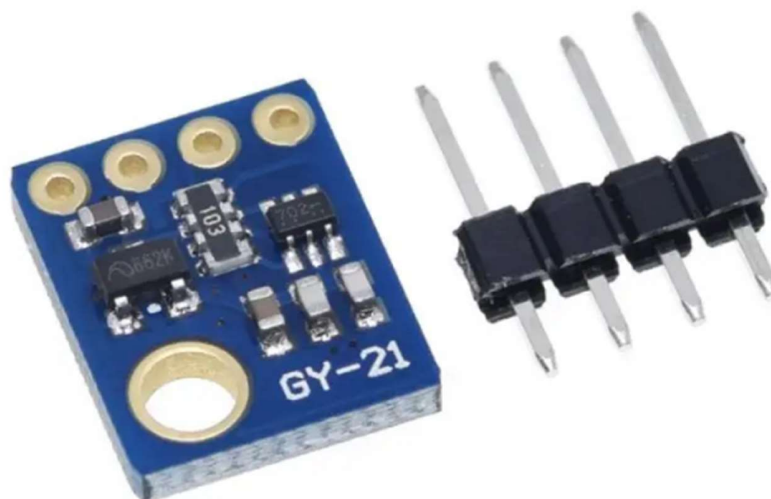


Рисунок 1.4 – Електронний датчик вологості та температури GY-21 HTU2

Електронні термометри відрізняються від інших пристроїв тим, що вони можуть генерувати вихідний сигнал, який може бути використаний для дистанційної індикації температури, запису, збереження або передачі даних. Термопари, напівпровідникові термометри або термометри та електричні елементи опору зазвичай використовують як чутливі елементи [9].

Електронний термометр використовує фізичні властивості провідника, щоб змінювати свій електричний опір відповідно до температури. Вимірювання передаються на світлодіодний або рідкокристалічний дисплей. Цей датчик може вимірювати температури в діапазоні від -50 до +100 градусів за Цельсієм. Можна забезпечити автономну роботу пристрою, підключивши батарею або джерело живлення зі стабільною напругою, якщо це необхідно [11].

Інфрачервоні термометри (рис. 1.5) відрізняються від інших типів термометрів які ми розглядали раніше, оскільки вони дозволяють вимірювати температуру без контакту з термометром.



Рисунок 1.5 – Інфрачервоний датчик температури

Інфрачервоний датчик який зображено на рисунку 1.5 може вимірювати температури від $-85\text{ }^{\circ}\text{C}$ до $4000\text{ }^{\circ}\text{C}$. Цей датчик визначає хвилю яку випромінює енергію через світлові та теплові промені від нагрітого тіла [12].

1.1.2 Розвиток гігрометрів

Рівень вологості визначається кількістю водяної пари в повітрі. Ця пара виникає, завдяки випаровування води з водойм, це можуть бути річки, озера або навіть поверхня землі. Багато аспектів нашого життя залежать від рівня вологості.

Першим фактором є те, що вологе повітря може впливати на погоду. Утворення хмар і опадів є результатом високого рівня водяної пари в повітрі, що

впливає на погодні умови, такі як дощ, сніг, туман і т. д. В той же час суха та сонячна погода часто супроводжується низькою вологістю.

По-друге, на здоров'я людини впливає рівень вологості. Із за високої вологості сприйняття тепла сприймається більш неприємним, оскільки шкіра не виводить піт в необхідні кількості, що призводить до дискомфорту. Навпаки, низька вологість може призвести до пересушування слизових оболонок і шкіри.

По-третє, на ріст і розвиток рослин впливає рівень вологості. Низька вологість може обмежити ріст і розвиток рослин, оскільки вони потребують води для фотосинтезу та життєдіяльності. Натомість надмірна вологість також може спричинити гнильну та захворювання рослин.

По-четверте, будівництво будівель залежить від рівня вологості. Вологість може вплинути на якість будівельних матеріалів, таких як дерево, метал і бетон, руйнуючи та пошкоджуючи їх.

Таким чином, вимірювання вологості є життєво важливими для багатьох сфер нашого життя та навколишнього середовища [14].

Ніколас да Куза винайшов перший пристрій для вимірювання вологості. Ця потреба виникла під час торгівлі вовною, оскільки вага вовни змінювалася залежно від погодних умов. Торговці помітили, що однакова кількість вовни може коштувати дорожче в дощі, ніж в ясній погоді. Отже, вони очікували дощу, а покупці очікували сухого повітря та погоди.

Ніколас да Куза розробив перший пристрій для вимірювання вологості, щоб точно визначити вагу вовни незалежно від погоди. Пристрій складався з двох чаш для вимірювання. На одну з них поклали більше вовни, а на іншу — каміння такої ж ваги. Під час торгівлі Ніколас додавав каміння на чашу терезів кожного разу, поки не міг встановити рівновагу. Після цього він підрахував кількість іншого каміння, щоб дізнатися, скільки рідини було у вовні.

У результаті цього винаходу кожен покупець міг перевірити кількість рідини, що міститься у вовні, і торговці могли розрахувати ціну на вовну на основі результатів вимірювань [15].

Після цього Леонардо да Вінчі створив перший пристрій для вимірювання вологості сировини, який називався гігрометром Леонардо. Ця конструкція пристрою була надзвичайно проста. Леонардовий гігрометр складався з двох ваг, на які клали віск і вату в рівних кількостях, як показано на рисунку 1.9.



Рисунок 1.6 – Гігрометр Леонардо да Вінчі

Після цього німецький математик Іоганн Генріх Ламберт запропонував використовувати пристрій для вимірювання вологості під назвою «гідрометр», який пізніше став «гігрометром». У 1774 році Ламберт розробив механізм, який можна було виміряти за допомогою натуральної шкіри. Крім того, він став першою людиною, яка виявила зв'язок між температурою та відносною вологістю.

У 1783 році було створено Десауссурой волосяний гігрометр, який використовував людське волосся для вимірювання вологості (рис.1.10). Вважалося,

що для вимірювання краще підходить світле волосся. Тим не менш, через природні відмінності людського волосся волосяні гігromетри показують похибку від $\pm$ до 2,5% [17].

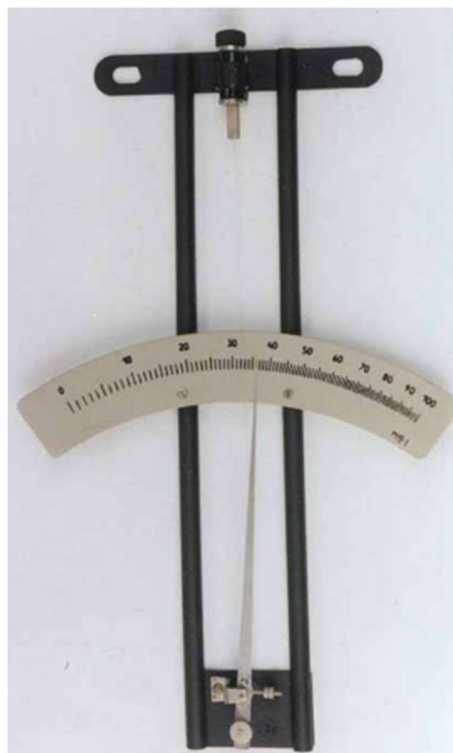


Рисунок 1.7 – Волосяний гігromетр

Для вимірювання вологості також існують сучасні гігromетри. Датчики контролю вологості використовуються в побуті для контролю мікроклімату, на підприємствах для забезпечення точності технологічних процедур і збереження обладнання, а також у сільськогосподарському секторі для оцінки якості ґрунту та його родючості. Налаштування кімнатних датчиків очевидно відрізняється від промислових. Крім того, існує різниця в методі вимірювання. Важливо знати, які вимірювання маються на увазі, для отримання результатів або встановлення пристроїв, які дозволяють працювати групами. Крім того, існує кілька потенційних варіантів:

- абсолютне значення в грамах на кубометр;
- середнє значення в одиницях RH (середня вологість);
- відсоток від маси досліджуваних зразків (тверді тіла та матеріали);
- і частина води на один мільйон частин ваги зразка, також відома як ppm (частин на мільйон).

Так, абсолютна вологість може коливатися від нуля до сто відсотків, що означає весь діапазон від повного насичення повітря до відсутності водяної пари. Більшість домашніх гігрометрів розроблені для оцінки абсолютної вологості [19].

Для вимірювання відносної вологості в промислових умовах частіше використовуються гігрометри, для цього вони використовують середнє значення в одиницях RH. Цей тип пристроїв зазвичай має вбудовані перетворювачі сигналів і легко підключається до відповідної вимірювальної системи. Багато з них також оснащені датчиками температури які вже вбудовані, це дозволяє ретельно контролювати внутрішній мікроклімат і завдяки цьому визначати зв'язок між рівнями температури та вологості в реальному часі. Це важливе для промислових операцій і досліджень, оскільки обидва ці показники можуть впливати на якість і ефективність роботи матеріалів і обладнання.

Психрометричні, аспіраційні, резистивні, ємнісні та інші види датчиків можна використовувати для вимірювання відносної вологості повітря.

1. Психрометричні датчики використовують метод охолодження повітря. Вони використовують вологий бульбаш для оцінки температур сухого та охолодженого повітря. Цими показниками можна визначити відносну вологість.

Психрометр складається з двох термометрів: один має суху поверхню, а інший — вологу; обидва термометри захоплені вологою тканиною, один край якої занурений у ємність із водою. Важливо відзначити, що тип термометра не має значення, чи вони спиртові чи рутні.

Психрометр використовується для вимірювання вологості тому, що чим менша вологість у повітрі, тим краще процес випаровування води з вологої поверхні термометра. Під час вимірювань обгорнутий термометр охолоджується, а сухий термометр продовжує фіксувати температуру повітря. Різниця між їхніми показниками вказує на рівень вологості. Цей рівень можна визначити за допомогою спеціальної психрометричної таблиці або за допомогою відповідної формули.

2. Аспіраційні датчики вимірюють вологість за допомогою масової та об'ємної концентрацій водяної пари. Вони підходять для автоматизованих систем контролю мікроклімату. Використовуючи принцип аспірації, детектори спостерігають за примусовим збором повітря за допомогою високочутливих лазерних датчиків. Такі пристрої надійно захищають місця, де встановлення стаціонарних пожежних сповіщувачів неможливо.

3. Ємнісні датчики оцінюють зміну ємності в залежності від зміни вологості. Зазвичай вони досить точні та мають широкий діапазон вимірювань. Ємнісні датчики належать до категорії датчиків положення, які також відомі як безконтактні вимикачі. Ці датчики реагують на зміни електричного поля в активній зоні виявлення, яка знаходиться поблизу електрода датчика. Це дозволяє ємнісним датчикам знаходити різні середовища навіть через обвідні труби чи неметалеві стінки контейнерів. Таким чином, вони часто використовують для визначення рівнів різних речовин. Датчик відчуває збільшення ємності середовища, що викликає відповідний сигнал. Ємнісні датчики добре оцінюють рідини (наприклад, воду чи паливо) і тверді матеріали (наприклад, деревину чи гранулят). Їх універсальність дозволяє їм працювати в багатьох галузях і сферах бізнесу.

4. Резистивні датчики оцінюють зміну опору матеріалу в результаті зміни вологості. Вони дають точні вимірювання та досить поширені. Датчики вологості на основі резистивного принципу вимірюють електричний опір солей. Опір електродів у сольовому середовищі змінюється зі зміною вологості повітря. Резистивний датчик вологості вимірює відносну вологість, враховуючи зміну

опору на основі вологості навколишнього середовища. Більшість датчиків вологості є електролітами, полімерами або датчиками з оксиду металу.

Ємнісний датчик вологості визначає вологість, вимірюючи зміну ємності між двома детектуючими електродами на напівпровідниковій підкладці. Використання ємнісного датчика вимірює вологість шляхом кореляції зміни електростатичної ємності елемента з вологістю навколишнього середовища.

Вибір певного типу датчика залежить від певних умов і потреб, а саме де він буде застосовуватися.

Так, датчики ємнісного і резистивного типу часто використовуються в офісних системах, які контролюють рівень вологості в діапазоні від тридцяти до сімдесяти відсотків в приміщенні.

Датчики вологості ємнісного типу працюють за допомогою конденсатора з повітрям як діелектрика в зазорі (рис. 1.11). Зміна вологості діелектрика змінює ємність повітряного конденсатора. Оскільки діелектрична проникність повітря пов'язана з вологістю, ємність конденсатора змінюється відповідно до вологості. Вимірювання та інтерпретація цієї зміни ємності дозволяють визначити рівень вологості повітря.

Цей принцип роботи дозволяє точно вимірювати вологість у діапазоні відносно низьких до середніх значень, які є типовими для офісних приміщень та домашніх приміщень.

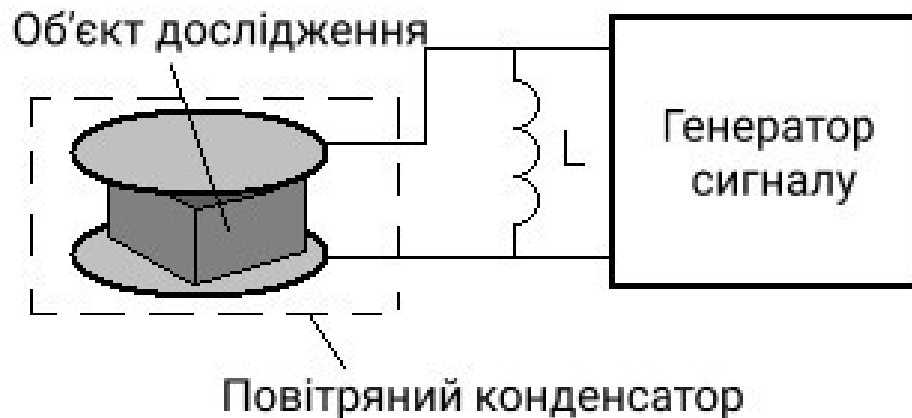


Рисунок 1.8 – Схема емнісного датчика вологості

Використання емнісних або резистивних датчиків вологості може бути недоцільно в агропромислових комплексах, таких як теплиці, грибні виробництва та овочесховища, де вологість може бути підвищеною та може відбуватися конденсація. Цей тип датчиків, особливо при високих рівнях вологості та конденсації, може показати значення з незначною похибкою.

Психрометричні датчики вологості, спеціалізовані для вимірювання вологості в умовах конденсації та високої вологості, рекомендуються для використання в таких умовах (рис. 1.9). Психрометричні датчики вимірюють вологість за допомогою мокрого термометра. Навіть у умовах високої вологості цей метод дозволяє точно визначати вологість повітря.

Завдяки використанню психрометричних датчиків в агропромислових комплексах можливо забезпечити точні та надійні вимірювання вологості, що є важливим для агропромислових процесів і вирощування рослин у нестандартних умовах [21].

Таким чином, для вимірювання вологості в умовах повітряного потоку потрібні аспіраційні (психрометричні) датчики, які також потребують вентилятора. Коли вентилятор працює, він допомагає створити нормований повітряний потік, що дозволяє робити більш точні вимірювання вологості, навіть коли відносна вологість повітря висока. При використанні аспіраційних датчиків які доповнені

вентиляторами можливо знизити похибку вимірювань до 1%. Завдяки подібним приладам можливо отримувати точні вимірювання в приміщеннях де є високий рівень вологості, оскільки інші види датчиків можуть давати менш точні результати.

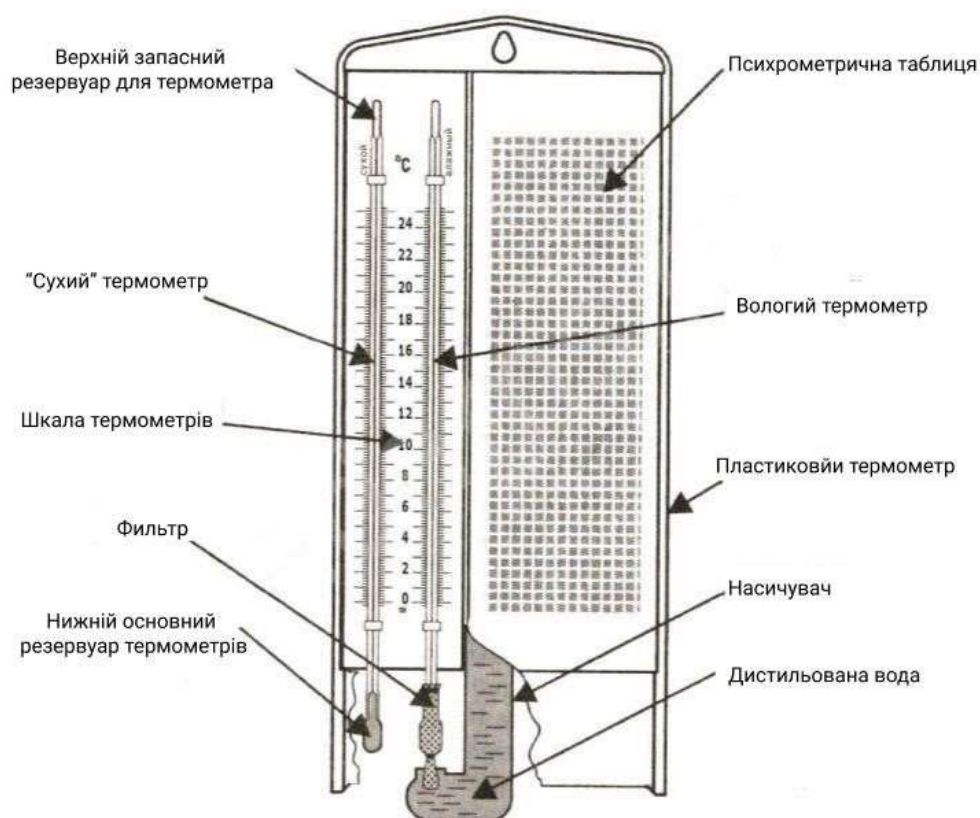


Рисунок 1.9 – Психрометричний датчик вологості

Дійсно, датчики вологості повітря мають широкий спектр застосування та відіграють важливу роль у багатьох галузях економіки. Ось кілька прикладів використання датчиків вологості:

1. Промисловість: підтримка мікроклімату в виробничих приміщеннях, де контроль вологості важливий для ефективної роботи чутливого обладнання.
2. Офісні приміщення: для забезпечення комфорту та здоров'я працівників, це забезпечується завдяки вимірюванню та регулюванню рівня вологості.

3. Жилово-комунальне господарство (ЖКГ): Контроль вологості в котельнях і водоочисних станціях, щоб зменшити конденсацію.

4. Будівництво: регулярний контроль вологості для запобігання розвитку грибків і цвілі на стінах і складах будівлі.

Ці пристрої забезпечують зручність і безпеку в різних сферах бізнесу та життя, а також допомагають підтримувати найкращі умови в різних середовищах.

1.1.3 Розвиток Барометрів

Наше спадкоємство в галузі науки і технології завжди включало надання історичним постатям заслуг за розвиток наукових ідей і створення приладів. Евангеліста Торрічеллі та Галілео Галілей відіграли важливу роль у розвитку атмосферних досліджень, а також у розробці барометра, який міг виміряти атмосферний тиск. За допомогою свого ртутного барометра Торрічеллі був першим, хто довів існування атмосферного тиску (Рис. 1.10).

Термін «барометр» походить від грецьких слів «барос», що означає «тяжість», і «метрос», що означає «вимірювач тиску». Ртутний барометр є важливим кроком у розвитку метеорології та особливо у дослідженні погодних умов, оскільки він дозволив вимірювати тиск повітря, а це в свою чергу стало однією із основних характеристик для прогнозування погоди [24].

Барометр Торрічеллі

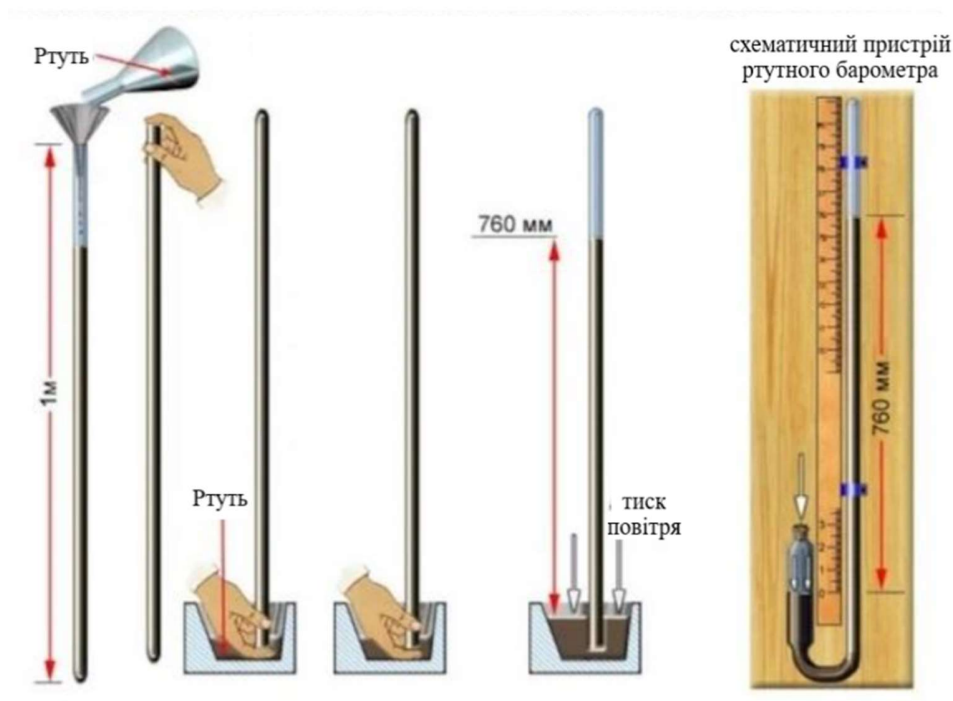


Рисунок 1.10 – Барометр Торрічеллі

Одним із великих недоліків ртутних барометрів є те, що вони містять ртуть, яка є потенційно небезпечною для здоров'я людини. Ртуть - важкий метал, який у невеликих кількостях може випаровуватися у вигляді шкідливих парів. Із за вдихання ртутних парів можливо отруїться ртуттю, що може призвести до серйозних проблем зі здоров'ям. Цей ризик обмежував використання ртутних барометрів у побуті або в місцях, де багато людей, щоб не було ризику пошкодження приладу.

У XVII столітті німецький вчений-математик Готфрід Вільгельм Лейбніц запропонував ідею створення анероїдного барометра, який не використовує ртуть. Анероїдний барометр, як іноді називають, є реальною альтернативою ртутному барометру і має багато переваг, зокрема те, що він зручніший у використанні вдома та безпечніший для здоров'я. Крім того, анероїдні барометри кращі, ніж ртутні барометри, оскільки вони не вимагають регулярного налаштування та обслуговування і такі барометри не використовують ртуть.

Анероїдний барометр використовує гофрований металевий циліндр, який в свою чергу реагує на зміну атмосферного тиску. Зміни форми циліндра виникають через зміни тиску, це в свою чергу впливає на покази приладу. У результаті цієї конструкції можна виміряти атмосферний тиск без використання ртуті або інших небезпечних матеріалів.

В 1847 році французький вчений Люсьєн Віді створив перший анероїдний барометр. Завдяки цьому анероїдні барометри стали широко використовуватися як у професійних додатках так і в побутових, зокрема в метеорології та аеронавтиці [26].

Барометр, як і раніше широко використовують в сучасному світі. Сучасні барометри можуть збирати дані про атмосферний тиск як на відкритому, так і в закритому середовищі. Авіація також широко використовує подібні пристрої для вимірювання висоти польоту над рівнем моря. Норма атмосферного тиску зазвичай становить 760 мм ртутного стовпа при температурі +15 градусів Цельсія.

Барометри бувають декількох видів:

1. Ртутні барометри.
2. Барометри електронного типу.
3. Анероїдні барометри, або механічні барометри.
4. Барометри рівня води [27].

Італійський фізик Евангелісто Торрічеллі винайшов ртутний барометр. Цей пристрій працює таким чином: довгу скляну трубку закривають одним кінцем, а потім її наповнюють рідким ртутним металом. Після цього трубку перевертають, а відкритий кінець вставляють у миску з ртуттю. Цей процес призводить до того, що невелика кількість ртутного металу виходить з трубки, залишаючи вгорі пустоту. У нормальних умовах висота ртутного стовпчика в скляній трубці становить приблизно 76 сантиметрів. Тиск повітря тисне на поверхню ртуті в мисці на дні барометра, тому трубка залишається в цьому стані. Тим не менш, вакуум у верхній частині скляної трубки майже ніколи не надає тиску на ртутний стовп. Таким

чином, висота ртутного стовпа в скляній трубці показує загальний тиск і атмосферу, які були присутні під час вимірювання.

Теоретично, барометр можна зробити з будь-якої рідини. Однак є кілька факторів, які впливають на вибір ртуті в якості робочої рідини. По-перше, щільність ртуті означає, що колона рідини під тиском повітря має невелику висоту. Порівняно з цим барометр, який використовує воду замість ртуті, повинен бути більше 100 метрів у висоту. Крім того, через дуже низький тиск пари ртуть майже не випаровується. У водному барометрі ситуація була б абсолютно іншою, оскільки вода має значно вищий тиск пари. З цієї причини важливо брати до уваги цей фактор у верхній частині барометра, хоча ртутний барометр майже не впливає на це [28].

Барометри, розроблені за методом Торрічеллі, вважаються дуже точними та корисними приладами для вимірювання тиску. З іншого боку, більшість людей, які шукають барометри для власного використання, віддають перевагу більш безпечним механічним барометрам, подібним анероїдним барометрам, які не містять ртуть. Анероїдні барометри містять датчик у закритій металевій коробці замість ртутного басейну, який реагує на тиск атмосфери (рис. 1.10).



Рисунок 1.10 – Барометр-анероїд мембранний метеорологічний (БАММ)

Металева коробка анероїдного барометра реагує на зміни тиску повітря. Залежно від тиску повітря вона або стискається, або розширюється, що впливає на позицію пружини, яка в свою чергу прикріплена до коробки. Відповідно, пружина взаємодіє з покажчиком на циферблаті. З зростанням тиску, коробка стискається, або навпаки втягується всередину. У результаті пружина розширюється, що призводить до того, що покажчик на циферблаті переміщується у напрямку годинникової стрілки. Звужуючи пружину, покажчик рухається проти годинникової стрілки, коли тиск падає, коробка розширюється або виступає назовні.

Для вимірювання тиску на анероїдному та отримання актуальної інформації необхідно барометр різко постукати по склу. Це змінить положення пружини та перемістить покажчик на циферблаті, щоб показати поточний тиск. Важливо пам'ятати, куди рухається стрілка. Якщо стрілка рухається вгору, це означає, що тиск зростає, і очікується стабільна та суха погода; якщо стрілка рухається вниз, тиск знижується, і це може привести до того, що погода може стати прохолоднішою та вологішою [29].

1.2 Приклади сучасних метеостанцій

Метеорологічні станції класифікують за різними критеріями. Станції можна класифікувати на такі чотири основні категорії:

- Дорожні;
- Ліси;
- гідрологічні станції;
- домашні метеостанції.

1.2.1. Дорожні метеостанції

Цей тип метеорологічних станцій створений для того, щоб забезпечити безпечну поїздку для водіїв та пасажирів.

Дорожні метеостанції складаються з наборів інструментів, призначених для збору, систематизації та аналізу метеорологічних даних. Дані форматуються та передаються до диспетчерського пункту, а також виводяться відповідні повідомлення на дорожні табло за допомогою спеціального програмного забезпечення.

Датчики інтенсивності та швидкості транспорту можуть збирати інформацію про напрямки руху транспорту, його кількість і класифікацію. Середня швидкість потоку, рівень завантаженості дороги, напрямок руху та смуга, а також інтервальність руху можна визначити за допомогою отриманих даних.

Двопроменеві та однопроменеві радары визначають середню швидкість транспортного потоку та швидкість кожного транспортного засобу. Обладнання з високою роздільною здатністю дозволяє отримати точні зображення кожного транспортного засобу. Кут огляду до 65°, щоб забезпечити найбільшу площу покриття за найменшу горизонтальну проекцію.

Ці станції зазвичай підключаються до табло, де відображається інформація про стан навколишньої температури повітря та поверхні дороги. Такі табло також можуть відображати попередження для водіїв під час руху, як-от: «Увага, мокра дорога / бічний вітер / ожеледиця і т.д.». Це допомагає водіям приймати розумні рішення та адаптувати свій стиль водіння згідно погодним умовам.

Дорожні метеостанції можуть вимірювати температуру поверхні дороги. Це все досягається за допомогою спеціальних датчиків, які знаходяться на глибині приблизно 30 см під поверхнею асфальту. Оскільки температура дорожньої поверхні може впливати на сцеплення шин з дорогою та загальну безпеку на дорозі, ці дані можуть стати життєво важливими для водіїв [30].

Таблиця 1.1 містить технічні характеристики сенсорів АДМ.

Таблиця 1.1 – Загальні технічні характеристики сенсорів АДМ.

Характеристика	Значення
Датчик температури та вологості повітря	
Діапазон вимірювання температури, °С	-40 - 50
Діапазон вимірювання вологості, %	10 - 98
Датчик опадів	
Класифікація типів опадів	Дощ, дощ з снігом, сніг, мокрий сніг
Розмір зафіксованих частинок атмосферних опадів, мм	1 – 5,0
Мінімальна кількість атмосферних опадів, мм	0,2
Датчик стану дорожнього покриття	
Діапазон вимірювань товщини шару води (при концентрації ПОМ не більше 0,1%), мм	0,1 – 4,00
Діапазон вимірювань товщини шару водного розчину NaCl, мм	0,1 – 2,00
Діапазон вимірювань концентрації водного розчину NaCl,%	0,2 – 20,00
Напруга живлення постійного струму, В	12 ± 0,6
Максимальна споживана потужність, Вт	0,8
Датчик температури поверхні і ґрунту	
Діапазон вимірювання температури, °С	-50 – 50
Датчик напрямку вітру	
Діапазон вимірювання напрямку вітру, °	0 - 356
Датчик швидкості вітру	
Діапазон вимірювання швидкості повітряного потоку, м/с	0,7 - 30
Датчик рівня води	
Діапазон вимірювання відстані до рідини, м	0,45 - 50

Для прикладу розглядається автономна дорожня метеостанція. (АДМ) «КОНДОР» (рис. 1.12).

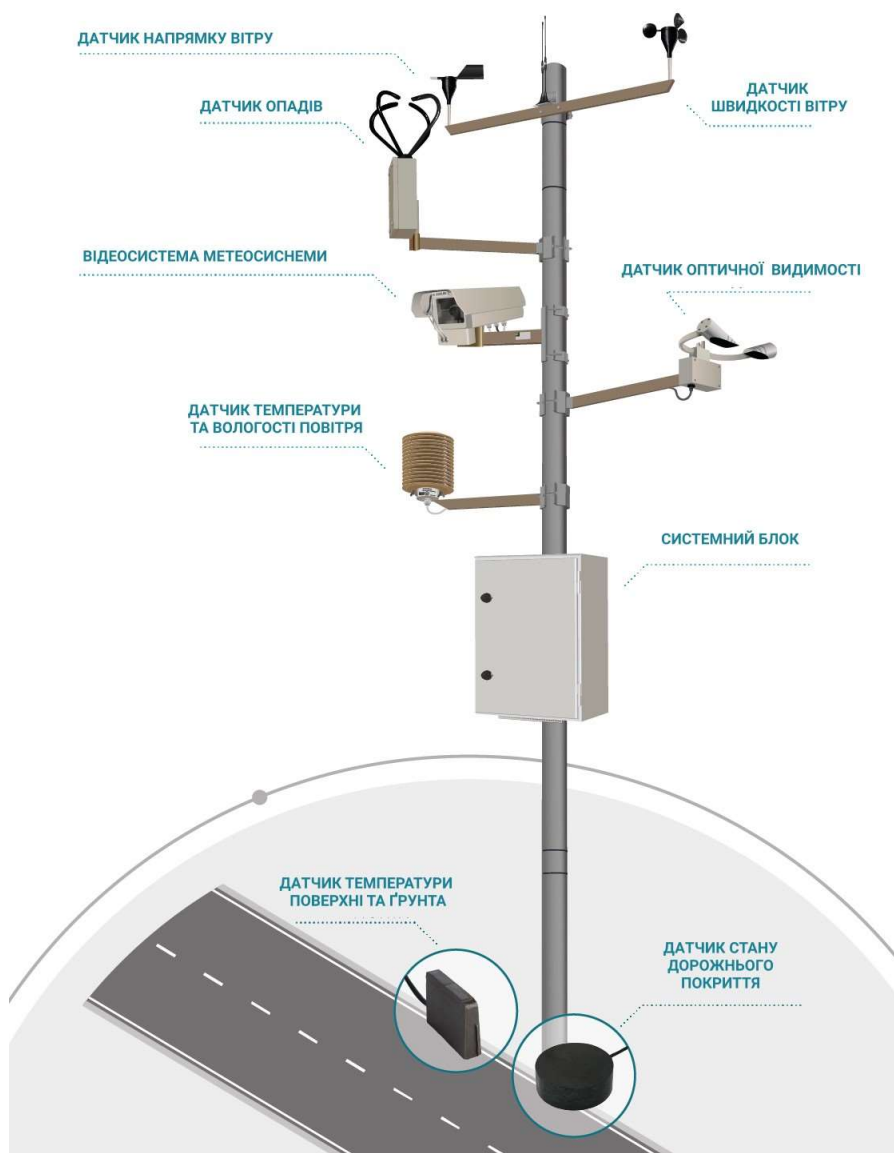


Рисунок 1.12 – АДМ «КОНДОР»

Дорожні служби можуть контролювати погодні умови та стан доріг за допомоги цієї станції, зокрема, щоб оперативно реагувати на негоду та несприятливі метеорологічні умови.

Для цієї автоматизованої метеорологічної станції (АДМ) основними завданнями є:

Такий тип автоматизованої метеорологічної станції (АМС) виконує наступні основні завдання:

1. Актуальна інформація про дороги та погоду доступна цілодобово.
2. Надання водіям інформації про стан доріг і погоду сприяє безпеці на дорогах.
3. Зменшення витрат на відновлення дорожнього покриття через передбачення негативних умов і уникнення їх наслідків.
4. Зменшення можливості прогнозування погоди та наявність системи раннього попередження, яка дозволяє дорожнім службам вживати заходи для забезпечення безпеки на дорогах і оперативно реагувати на погіршення погоди.

1.2.2 Метеорологічні станції для лісу (ЛМС)

Головною метою лісових метеостанцій в першу чергу є запобігання та швидке виявлення лісових пожеж. Так як метою цих станцій є забезпечення безпеки лісових територій від пожеж і вживання заходів для їх запобігання, тому такі станції виконують більше завдань, ніж традиційні метеорологічні спостереження. Метеостанцій розташовані в лісових масивах, дозволяють отримати важливі дані для вирішення цієї проблеми:

1. Щоб визначити пожежонебезпеку, оцінка рівня вологості дерев і ґрунту допомагає визначити, наскільки сухою або вологою є лісова рослинність і ґрунт.
2. Метеорологічні станції відстежують температуру на різних рівнях лісового масиву, щоб визначити, як змінюються пожежонебезпечні умови на різних висотах.

3. Дані які збираються використовуються для створення карт, які відображають найбільш пожежонебезпечні місця. Цей тип карт дозволяє зосередити увагу та ресурси на місцях, де існує найбільший ризик пожежі, і забезпечити більш ефективний нагляд і управління цими районами [32].

Компанія «Dyasop» пропонує компактне рішення лісових метеостанцій, яке має багато функцій, незважаючи на свої невеликі розміри (рис. 1.13). Цей пристрій може вимірювати:

1. температуру повітря,
2. тиск
3. швидкість вітру
4. рівень вологості повітря
5. температуру та вологість ґрунту
6. Рівень дощу.

У разі виникнення незвичайних погодних умов або ситуацій станція також має мобільний зв'язок, що дозволяє користувачам отримувати повідомлення на свої мобільні телефони. Крім того, користувачі можуть запитувати звіти про погоду, відправляючи до станції відповідні текстові команди.

Завдяки сучасним можливостям моніторингу та сповіщення про погоду ця метеостанція дозволяє оперативно реагувати на зміни у погодних умовах та довкіллі. Захист лісових ресурсів і забезпечення безпеки лісових територій є життєво важливими.

Легкість транспортування метеостанції з одного місця в інше є важливою перевагою. Це робить станцію більш мобільною та зручною в обслуговуванні, оскільки для розбирання та складання не потрібен спеціальний інженерний досвід. Швидке встановлення та розбирання станції робить її більш адаптивною до різноманітних умов і завдань, які можуть виникнути під час роботи.



Рисунок 1.13 – Метеостанція Dyacon MS-150

Таким чином, Dyacon MS-150 має такі функції:

- Метеостанція живиться від 12–24 В;
- Самостійна реєстрація даних;
- Підключення кабелю Modbus (Modbus — це відкритий протокол комунікацій, заснований на архітектурі master-slave). Часто використовується в промисловості для забезпечення зв'язку між електронними приладами) [34];
- Вимірює швидкість та напрямок вітру;
- Вимірює температуру повітря;
- Вимірює тиск;
- Вимірює відносне вологість;
- Має доступ до даних про кількість опадів;
- Вимірює температуру та вологість ґрунту;

- Має сонячні панелі [33].

1.2.3 Гідрологічні метеорологічні станції

Гідрологічні метеорологічні станції (ГМС) мають вирішальне значення для збереження водних ресурсів і прогнозування різних водних явищ. Вони відповідають за наступні обов'язки:

1. Збереження питної води: ГМС ведуть спостереження за рівнем водойм, які постачають питну воду населенню. Це допомагає гарантувати постійне постачання питної води та швидко виявлення будь-яких змін у кількості чи якості води.

2. Прогноз повеней та посух: За допомогою ГМС можливо прогнозування та вчасно виявляти небезпечні ситуації, подібні повені або посухи. Це в свою чергу, дозволяє запобігати шкоді та захистити життя та майно.

3. Моніторинг зсувів, ураганів і смерчів: ГМС спостерігають за погодою та природними явищами, такими як зсуви, урагани та смерчі, і інформують про важливу інформацію, яка захищає від цих стихійних подій.

Місцевості з водосховищами та водовідвідними мостами, а також місця, де часто виникають повені, мають високий попит на ГМС. Вони сприяють управлінню водними ресурсами та вжиття заходів для запобігання стихійним явищам [35].

Для забезпечення безпеки та попередження катастроф моніторинг та прогнозування екстремальних гідрометеорологічних подій є надзвичайно важливими. За допомогою супутникового дистанційного зондування можна отримати важливу інформацію для цілей, таких як спостереження гідрометеорологічних подій. За допомогою супутникового спостереження можна відстежувати зміни рівня водойм, визначати можливі наслідки та поширення повеней і визначати масштаби повеней. Це допомагає швидко приймати рішення

щодо захисти або евакуації населення та майна з територій де можлива повінь. Супутникові дані роблять можливим моніторинг екстремальних подій у всьому світі; це особливо важливо для передбачення та реагування на зміни клімату та навколишнього середовища, які відбуваються в глобальному масштабі. Про опади, топографію, рослинність і гідрографію, дистанційне зондування надає інформацію про фактори, які впливають на ці явища, і в випадку екстремальних події інформує про це оператора. Це допомагає визначити місця з підвищеним ризиком і розробити заходи для пом'якшення цих подій. У поєднанні з гідрологічним моделюванням супутникове дистанційне зондування є ефективним інструментом для прогнозування та швидкої реакції на надзвичайні гідрометеорологічні події, що може допомогти зменшити ризики для життя та майна [36].

Telemetry System SMART2000 – це сучасна гідрометрична станція, що збирає та передає дані про гідрологічні та погодні явища в режимі реального часу. SMART2000 також надає можливість відстежувати погоду та гідрологічні умови так місцях як водні басейни та інших подібних об'єктах.

Система, яка базується на мікропроцесорі ARM і має можливості керування Ні-Тес, може виконувати різноманітні завдання. Її можна використовувати в багатьох областях, таких як екологія, метеорологія, гідрологія та контроль якості повітря та води. Термістор Pt100 використовується для вимірювання температури. Для підключення використовується один із аналогових входів. Після цього аналогові входи використовуються для збору даних про швидкість, температуру, напрямок вітру, кількість опадів та рівень вологості.



Рисунок 1.14 – SMART2000 Telemetry System

Основні характеристики Telemetry System SMART2000 включають:

- низьке енергоспоживання;
- математичне поєднання каналів датчиків;
- Flash-пам'ят на 7 Мб;
- Цифрових входів - 6;
- Аналогові входів - 8;
- SD-карта пам'ять яка підтримує до 2 Гб;
- порт USB 2.0;
- Цифровий перетворювач 16-розрядний;
- Діапазон температур від -40 ° до +70°.

1.2.4. Метеостанції для домашнього користування

Домашні метеостанції - це невеликі пристрої для спостереження за погодою, оснащені різними типами датчиків. Основною метою цих станцій є вимірювання різних параметрів, таких як вологість, температура, тиск (в приміщенні та на вулиці), а також інших. Особливо корисно подібні метеостанції для власників дач і фермерів, оскільки ці професії значною мірою залежать від погоди [38].

Використовуючи систему Tempest, WeatherFlow (рис. 1.17) є одним із найкращих способів отримання прогнозів погоди в місці, де встановлений датчик цієї компанії.

У Tempest є багато переваг перед іншими домашніми метеостанціями, одним із таких особливостей є те, що система не тільки робить вимірювання параметрів навколишнього середовища, але ще проводить аналіз даних яка вона отримала у співставленні з даними від радарів і супутників інших метеостанцій. Система стандартизує всі дані з супутників і передає їх для аналізу в додаток, перш ніж надавати результати. Це необхідно для того, щоб забезпечити найвищу точності, а саме головне надійність результатів, оскільки зовнішні фактори можуть впливати на показники різними способами.

Додаток для системи Tempest розроблений таким чином, щоб було легко користуватися. Він надає доступ до понад п'ятнадцяти показників погоди, таких як швидкість вітру, температура і УФ-індекс. Крім того, що робить використання Tempest ще зручнішим, додаток дозволяє відстежувати результати вимірювань у вигляді діаграм і графіків.



Рисунок 1.15 – Метеостанція компанії WeatherFlow

Tempest System включає:

- Сенсор тиску та сенсор світла передають дані щодо тиску, сонячного випромінювання та рівня освітленості.
- Сонячні батареї;
- Безпроводний зв'язок, який працює на великій відстані від приміщення;
- Тактильний датчик дощу, який фіксує початок, інтенсивність та тривалість дощу;
- Сонячний сенсор дощу;
- Датчики вологості та температури;
- датчик освітлення, який може вимірювати освітлення на відстані до 40 км [39].

Висновки до розділу 1

У даному розділі було вивчено прогрес розробки термометрів, включаючи різноманітні види та принципи їх роботи. Освітлено етапи та еволюцію гігрометрів, де докладно розглянуто види та принципи роботи датчиків вологості повітря. Детально проаналізовано розвиток барометрів, включаючи різні види та основні принципи роботи датчиків атмосферного тиску.

Додатково, розглянуто сучасні метеостанції, охоплюючи різні типи, такі як дорожні метеостанції, лісові метеорологічні станції (ЛМС), гідрологічні метеорологічні станції та побутові домашні метеостанції. Детально висвітлено приклади функціональності та застосування кожного типу метеостанції, що дозволяє отримати широкий огляд сучасних досягнень в цій галузі.

У зв'язку з високою вартістю метеостанцій, які автоматично слідкують за станом клімату, було вирішено розробити систему автоматичного керування метеостанцією. Ця система забезпечує збір та аналіз метеорологічних даних, а також дозволяє користувачам отримувати актуальну інформацію про погодні умови.

У зв'язку з цим були поставлені наступні завдання:

- Розробити апаратну частину метеостанції, яка забезпечує збір метеорологічних даних.
- Розробити програмне забезпечення для збору та аналізу метеорологічних даних.
- Розробити веб-інтерфейс для користувачів, який дозволяє отримувати актуальну інформацію про погодні умови.

2 ОБҐРУНТУВАННЯ ТА ВИБІР КОМПОНЕНТІВ

2.1 Функціональна схема приладу

Основним компонентами в схемі є мікроконтролер, WiFi модуль, датчики: тиску, температури та вологості (рис. 2.1).



Рисунок 2.1 – Функціональна схема автоматизованої метеостанції з мікроконтролерним керуванням

На схемі зображено передачу аналогового сигналу сенсору, що надходить від потенціометра. Цей аналоговий сигнал передається в АЦП, де перетворює в цифровий та передає інформацію на мікроконтролер.

Завдяки wifi модулю можливе з'єднання до існуючої мережі Wi-Fi та створення веб-сервера. Коли будь-який підключений пристрій звертається до цього

веб-сервера, оброблені дані з датчиків передаються у веб-браузер цього пристрою у вигляді показників температури, тиску та вологості. При дистанційній роботі для живлення приладу буде використовуватись power bank.

На даному етапі запис, обробка та передача даних здійснюється завдяки програмному забезпеченню, що вшите в мікроконтролер.

2.2 Вимоги для елементної бази приладу

Для розробки автоматизованої метеостанції потрібно визначитися з ключовими параметрами майбутнього пристрою, які повинні відповідати таким вимогам:

- Точність сенсорів не перевищує ± 1 °C;
- Можливість з'єднання з комп'ютером через USB-вихід;
- Захисний корпус для виносного датчика температури;
- Можливість підключення до зовнішнього джерела живлення при дистанційній роботі пристрою;
- Робота при температурі нижче нуля;
- Підключення без використання додаткових перехідників;
- Компактні розміри;
- Можливість передачі даних по WiFi мережі;
- Доступність;
- Можливість подальшої модернізації пристрою.

2.2.1 Керувальний цифровий пристрій

При виборі мікроконтролера було визначено, що найбільш підходить WeMos D1 Mini на чипе ESP-12F ESP8266. Основні особливості, даного мікроконтролера є:

- Легкість використання. Благодаря простій мові програмування, обширному набору бібліотек і навчальних матеріалів, платформа стає доступною для всіх бажаючих.
- Доступність. Широкий асортимент елементів для проектування дозволяє обирати необхідний пристрій з різними параметрами. Для менш вимогливих або тренувальних проектів існує можливість придбати китайські аналоги, які часто не відстають в якості від оригіналу.
- Незалежність. Arduino підтримується різними програмами (ПЗ), такими як: Mac OS, Linux, Windows.
- Для порівняння обрано плати які само часто використовують і які мають модуль ESP8266: WeMos D1 mini , WeMos D1 на чіпі ESP-12F, Witty Cloud на чіпі ESP-12F, NodeMCU v3 на базі ESP-12E. Технічні характеристики плат приведені в табл.2.1

Таблиця 2.1 – Технічні характеристики плат

Характеристика	WeMos D1 mini	WeMos D1 на чіпі ESP-12F	Witty Cloud на чіпі ESP-12F	NodeMCU v3 на базі ESP-12E
Підтримувані мережі	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Частота мікроконтролера	80 МГц	80 МГц	80 МГц	80 МГц
Швидкість UART	115200	115200	115200	9600
робоча напруга	3.3 В	3.3 У	3.3 В	3.3 У
Напруга живлення	3.7-12 В	9-24 В	3.7-12 В	3.7-15 В
Максимальний струм споживання (при передачі)	240 мА	240 мА	240 мА	240 мА
Розміри	33 x 26x 6 мм	68.5 x 54 x 12 мм	33 x 26x 6 мм	59 x 31 мм

Виходячи з параметрів, які представлені у таблиці 2.1 можна зробити висновок, що доцільнішим буде вибрати плату WeMos D1 mini. З порівняні з іншими вона має менші розміри, меншу напругу живлення.

Плата WeMos D1 mini (рис.2.2) представляє собою аналог плат Witty Cloud і NodeMCU v3 з найменшими розмірами, використовуючи модуль ESP8266 у версії ESP-12F. Для функціонування WeMos D1 mini не потрібен зовнішній мікроконтролер чи інший управляючий пристрій, оскільки, крім Wi-Fi модуля ESP-12F, вже вбудовано 32-бітний мікроконтролер з тактовою частотою 80 МГц та флеш-пам'ять об'ємом 4МБ.

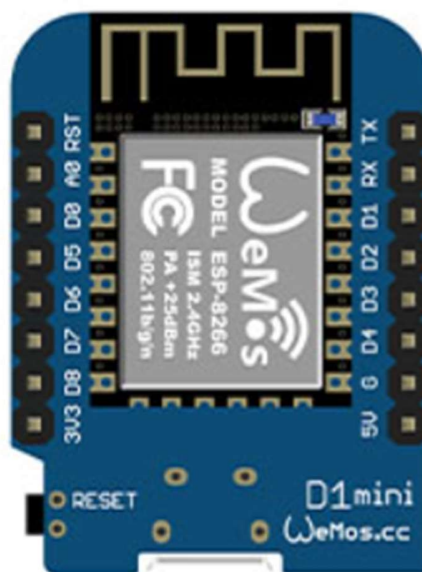


Рисунок 2.2 – Плата WeMos D1 mini

У самого чіпа esp8266 всі виходи мають числові позначення. На схемі їх позначено як GPIO_n, де n – номер. На платах (NodeMCU, Wemos Mini) піни мають позначення D_n, і ці номери не відповідають номерам GPIO! При використанні,

наприклад, з Wemos можна використовувати як номери висновків GPIO (`digitalWrite(5, LOW)`), так і D-позначення пінів на платі (`digitalWrite(D1, LOW)`) – якщо вибрано плату Wemos. Тут важливо бути уважним. Також GPIO1 та GPIO3 на платі позначені як TX та RX, і ви можете звертатися до них за цими назвами (`digitalWrite(TX, LOW)`).

Esp8266 розполагає великою кількістю системних пінів, і важливо бути особливо уважним при їх використанні. Деякі з цих пінів призначені для зовнішньої Flash-пам'яті, і в загальному випадку їх використання не рекомендується (деталі можна знайти в документації). На платі NodeMCU кількість візуально доступних пінів значно більша, ніж на Wemos Mini, проте щодо безпечного використання, їх кількість однакова. Також слід зазначити, що деякі з невикористаних пінів вимагають певного логічного рівня при включенні мікроконтролера (під час подачі живлення або перезавантаження). При підключенні до цих пінів чого-небудь із протилежним сигналом esp8266 може не запуститися. На платах, таких як NodeMCU, Wemos, і інші, вказані піни вже мають підтягнуті резистори до необхідного рівня напруги. Тому важливо обдумати, до чого саме ви підключаєте ці піни і як це може вплинути на напругу на піні. Наприклад, при підключенні енкодера, який заземлить системний пін до GND, esp може не запуститися.

Пін GPIO16/D0 не призначено для підключення переривань (`attachInterrupt()`) і генерації ШИМ сигналу (`analogWrite()`). Максимальний струм з GPIO - 12 мА. Світлодіод LED_BUILTIN розташований на піні GPIO2, і його робота інвертована: при подачі LOW він вмикається, і навпаки.

При старті контролера майже всі піни здійснюють перехід до високого рівня, в деяких випадках це може бути важливим. Єдиними "спокійними" пінами є D1 (GPIO5) та D2 (GPIO4). Якщо контролер управляє пристроями безпосередньо (реле, транзистор або інші пристрої), то краще використовувати саме ці піни!

На цих пінах розташована шина I2C, але її можна змінити на будь-які інші піни за допомогою `Wire.begin(sda, scl)`.

Також є свої особливості використання WiFi модуля. WiFi працює в синхронному режимі, і його обробник повинен систематично викликатися під час виконання програми не рідше, ніж кожні 20 мс, якщо WiFi використовується у програмі. Обробка WiFi відбувається в наступних точках:

1. Автоматично наприкінці кожної ітерації `loop()`.
2. У межах будь-якого `delay()`.
3. У середині функції `yield()`.

Якщо ваша програма містить участки, які виконуються тривалий час, слід розміщувати виклики `yield()` перед і після важких блоків коду. Також можна зустріти використання `delay(0)` у чужих кодах, що, по суті, є еквівалентом виклику `yield()`.

З тих же причин не рекомендується використовувати затримку `delayMicroseconds()` більше ніж на 20,000 мкс [40].

2.2.2 Датчики

Для порівняння було обрано три датчика: BMP280, BMP180, BMP085 (табл.2.2). На ринку сенсорів вони відрізняються простотою у використанні та бюджетною ціною. Головним критерієм було, щоб модуль в собі поєднував датчик атмосферного тиску, термометр і датчик вологості повітря.

Таблиця 2.2 – Технічні характеристики датчиків тиску.

Характеристики	BMP280	BMP180	BMP085
Діапазон вимірів	300...1100hPa (-500...9000м над ур. моря)	300...1100hPa (-500...9000м над ур. моря)	300...1100hPa (-500...9000м над ур. моря)
Точність (тиск)	1hPa	0.02hPa	0.02hPa
Напруга живлення:	1.8 - 3.6	1.8 - 3.6 В	1.8 - 3.6 В
Споживаний струм	2.7 мкА (при частоті опитування 1 Гц)	5 мкА (1 Гц).	5 мкА (1 Гц).
Тип показань	цифровий вихід I2C	цифровой выход I2C	цифровой выход I2C
Робочий температурний діапазон	-40°C..+85°C	от 0 до +65 °C	Відсутній
Датчик вологості	Присутній	Відсутній	Відсутній
Розміри	15x11 мм	21x18 мм	16,5 x 16,5 мм

Розглянувши характеристики декількох сенсорів було прийнято рішення обрати модуль BMP280. Оскільки він цифровий датчик тиску нового покоління, відомий своєю вдосконаленою конструкцією, відрізняється від попередніх моделей, таких як BMP085 та BMP180. Його основні переваги полягають у компактних розмірах, ефективному використанні енергії, можливості вимірювання вологості, високій роздільній здатності для показань тиску і температури, а також вбудованих фільтрах, які ефективно зменшують вплив шумів на отримані вимірювання датчиків.

Розглянемо особливості BMP280. Цей датчик використовує перевірену технологію п'єзорезистивного датчика тиску від компанії Bosch, що відзначається високою стійкістю до електромагнітної сумісності, високою точністю, лінійністю та довгостроковою стабільністю. Це абсолютний барометричний тисковий датчик, спеціально розроблений для мобільних пристроїв і вбудований в надзвичайно компактний корпус розміром лише $2,0 \times 2,5 \text{ мм}^2$ та висотою у 0,95 мм. Малі розміри та низьке енергоспоживання (2,7 мкА при 1 Гц) дозволяють використовувати його в батарейно-живлених пристроях, таких як мобільні телефони, GPS-модулі або годинники. Як наступник широко використовуваного BMP180, BMP280 надає високу продуктивність в різних застосуваннях, що вимагають точного вимірювання тиску. BMP280 працює з меншим рівнем шуму, підтримує нові режими фільтрації та має інтерфейс SPI на 63% менший, ніж BMP180.

Функціональний опис BMP280 включає в себе п'єзорезистивний чутливий елемент тиску та ASIC зі змішаним сигналом. ASIC відповідає за аналого-цифрове перетворення та передачу результатів та особливостей датчика через цифровий інтерфейс. Цей датчик надає велику гнучкість для дизайнера, оскільки може бути налаштований з урахуванням точності, часу вимірювання та споживання електроенергії. BMP280 пропонує три режими потужності: режим сну, нормальний режим та форсований режим. Режим сну відсутність вимірювань, нормальний режим включає автоматизоване перемикання між активним та неактивним періодами, а форсований режим передбачає одноразове вимірювання. Налаштування передискретизації дозволяє вибрати оптимальні параметри для точного вимірювання тиску та температури при різних обставинах. Датчик також обладнаний вбудованим IIR-фільтром для мінімізації короточасних перешкод у вихідних даних. Щоб спростити використання пристрою та зменшити можливі комбінації налаштувань, Bosch Sensortec пропонує рекомендації для типових випадків використання, таких як смартфони, метеостанції, польоти іграшок та інші.

На рис.2.3 зображена спрощена блок-схема BMP280:

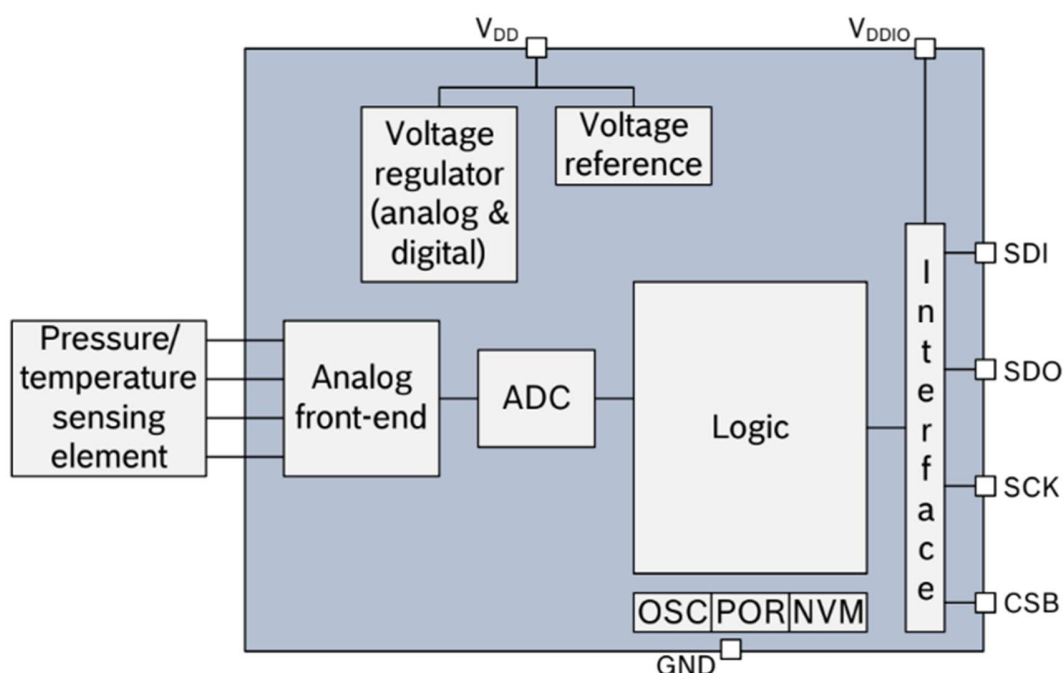


Рисунок 2.3 – Блок-схема BMP280

BMP280 розподіляє живлення через два окремі контакти:

- VDD є основним джерелом живлення для всіх внутрішніх аналогових і цифрових функціональних блоків.
- VDDIO — це окремий контакт живлення, який використовується для живлення цифрового інтерфейсу.

Логічна схема та значення регістру скидають генератор скидання, коли він працює. Нема обмежень щодо послідовності підвищення VDD і нахилу VDDIO. Датчик переходить у сплячий режим, коли живлення включено.

ВАЖЛИВО: Коли VDDIO вимкнено, утримання будь-якого контакту інтерфейсу (SDI, SDO, SCK або CSB) на логічному високому рівні може остаточно пошкодити пристрій через надмірний струм, що проходить через діоди захисту від електростатичного розряду. Якщо VDDIO подається, а VDD ні, контакти інтерфейсу зберігаються на високому рівні Z.

Період вимірювання BMP280 включає в себе здійснення вимірювання як температури, так і тиску, а також можливість вибору передискретизації. Після завершення цього періоду вимірювання дані передаються через додатковий фільтр

ІІР, який призначений для ефективного вилучення короточасних коливань тиску, наприклад, викликаних ударом дверей.

Принцип вимірювання тиску. Можливо включити або пропустити вимірювання тиску за допомогою BMP280. Пропуск вимірювання може бути корисним, коли BMP280 використовується лише для вимірювання температури. У випадку увімкнення вимірювання тиску доступні різні опції передискретизації. Кожен рівень передискретизації зменшує шум і збільшує вихідну роздільну здатність на один біт, тобто зберігається в регістрі даних XLSB 0xF9. Увімкнення або вимкнення вимірювання та налаштування передискретизації визначаються за допомогою бітів `osrs_p[2:0]` у регістрі управління 0xF4.

Вимірювання температури можна увімкнути або пропустити за бажанням. Пропуск вимірювання температури може бути корисним для швидкого вимірювання тиску. При включенні температурного вимірювання доступні різні параметри передискретизації. За кожен крок передискретизації шум зменшується, а роздільна здатність збільшується на один біт, і ця інформація зберігається в регістрі даних XLSB 0xFC. Увімкнення або вимкнення температурного вимірювання та налаштування передискретизації визначаються за допомогою бітів `osrs_t[2:0]` у регістрі управління 0xF4.

Для смягчення короточасних впливів, таких як удар дверей чи вікна або вплив вітру на датчик, BMP280 використовує внутрішній фільтр ІІР. Цей фільтр призначений для підглажування вихідних даних, зменшуючи ефект шумів, при цьому не викликаючи додаткового навантаження на інтерфейс та процесор. В результаті кожного кроку вимірювання вихідний сигнал проходить через фільтр, розрахований за певною формулою:

$$data_filtered = \frac{data_filtered_old \cdot (filter_coefficient - 1) + data_ADC}{filter_coefficient} \quad (2.1)$$

де `data_filtered_old` — це дані, отримані з попереднього збору, а `data_ADC` — це дані, які надходять з ADC до фільтрації IIR.

Рівень шуму залежить від параметрів передискретизації та обраних налаштувань фільтра. Вказані значення були встановлені в умовах з контрольованим тиском і ґрунтуються на середньому стандартному відхиленні 32 послідовних точок вимірювань, отриманих з найвищою швидкістю відбору проб. Це важливо для того, щоб виключити вплив довгострокових дрейфів на вимірювання шуму.

Нижче наведені підтримувані переходи між режимами (Рис. 2.4). У разі, якщо пристрій в даний момент проводить вимірювання, виконання команд перемикання режимів відкладається до завершення поточного періоду вимірювання. Подальші команди зміни режиму ігноруються до виконання останньої команди зміни режиму. Перемикання режимів, окрім тих, що показані нижче, було перевірено на стабільність, але не відповідає рекомендованому використанню пристрою.

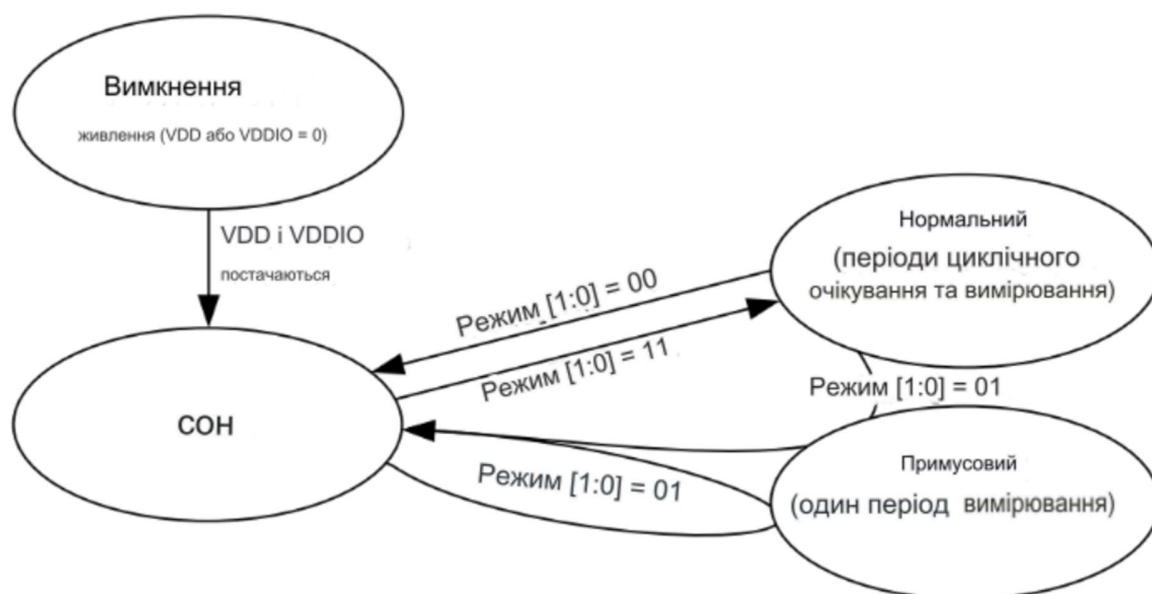


Рисунок 2.4 – Діаграма переходу режиму

Для отримання даних після проведення перетворення настійно рекомендується використовувати пакетне зчитування і уникати окремого звертання до кожного регістра. Це допоможе уникнути можливого переплутування байтів, які відносяться до різних вимірювань, і зменшить обсяг трафіку інтерфейсу. Для зчитування даних використовується пакетне зчитування від 0xF7 до 0xFC. Дані зчитуються у форматі беззнакового 20-бітного числа як для тиску, так і для температури. Для виконання зчитування та компенсації настійно рекомендується скористатися API BMP280, яке надається Bosch Sensortec.

Час для отримання даних у примусовому режимі слід визначити так, щоб забезпечити дотримання максимального часу вимірювання. У звичайному режимі зчитування може проводитися із швидкістю, схожою на очікувану швидкість виведення даних. Після отримання значень "ut" і "ur" необхідно розрахувати реальний тиск і температуру за допомогою параметрів компенсації, які зберігаються в пристрої.

У режимі звичайного вимірювання час вимірювання не обов'язково синхронізується із зчитуванням. Це означає, що нові результати вимірювань можуть стати доступні, навіть коли користувач вже читає результати попереднього вимірювання. Для забезпечення консистентності даних у такому випадку використовується затінення. Затінення працює ефективно тільки в разі, якщо всі регістри даних зчитуються в рамках одного пакетного зчитування. Тому важливо використовувати пакетне зчитування, якщо читання даних не синхронізується з циклом вимірювання.

Використання кількох незалежних команд читання може призвести до неузгоджених даних. Якщо нове вимірювання завершено, а регістри даних все ще зчитуються, нові результати вимірювань переносяться в тіньові регістри даних. Вміст тіньових регістрів передається в регістри даних, як тільки користувач завершує пакетне читання, навіть якщо не всі регістри даних були прочитані. Таким чином, зчитування в кількох регістрах даних може бути гарантовано послідовним

лише в межах одного циклу вимірювання, якщо використовується одна команда пакетного зчитування. Кінець пакетного зчитування вказується наростаючим фронтом виводу CSB у випадку SPI або визначенням стану зупинки у випадку I2C. Після завершення пакетного читання всі регістри даних користувача одразу оновлюються.

Вихідний сигнал BMP280 представлений значеннями аналого-цифрового перетворювача (АЦП). Однак, оскільки кожен чутливий елемент має унікальну характеристику, для отримання фактичних значень тиску та температури необхідно використовувати комплексні параметри калібрування. Рекомендований метод обчислення використовує арифметику з фіксованою комою. У високорівневих мовах, таких як Matlab™ або LabVIEW™, код із фіксованою комою може не працювати належним чином. В такому випадку можна використовувати код із плаваючою комою як альтернативу. Також, для 8-розрядних мікроконтролерів, слід враховувати можливі обмеження розміру змінної.

BMP280 підтримує цифрові інтерфейси I2C і SPI і виступає як підлеглий пристрій для обох зазначених протоколів. Інтерфейс I2C підтримує режими Standard, Fast і High Speed. Інтерфейс SPI, у свою чергу, підтримує як режим SPI «00» (CPOL = CPHA = «0»), так і режим «11» (CPOL = CPHA = «1») у 4- та 3-провідній конфігурації. Виконувані транзакції включають:

- Однобайтовий запис
- Багатобайтовий запис (з використанням пари адреси реєстру та даних реєстру)
- Однобайтове читання
- Багатобайтове читання (з використанням адреси одного регістра, яка автоматично збільшується).

Визначення інтерфейсу здійснюється автоматично на основі стану CSB (вибір мікросхеми). Коли CSB підключено до VDDIO, інтерфейс I2C стає активним. В разі опускання CSB інтерфейс SPI активується. Після одного витягнення CSB

(незалежно від того, чи відбувся будь-який тактовий цикл), інтерфейс I2C вимикається до наступного скидання живлення. Це здійснюється для уникнення ненавмисного декодування трафіку SPI як даних I2C. Так як скидання виконується лише тоді, коли VDD і VDDIO встановлені, ризику невірної декодування протоколу через послідовність увімкнення не існує. Однак, у випадку використання I2C і CSB підключений не безпосередньо до VDDIO через програмований контакт, важливо впевнитися, що цей контакт вже виводить рівень VDDIO під час скидання живлення пристрою. В іншому випадку пристрій буде заблоковано в режимі SPI і не буде реагувати на команди I2C.

Інтерфейс I2C у підлеглому пристрої сумісний із версією 2.1 специфікації Philips I2C. Підтримуються всі режими (стандартний, швидкий, високошвидкісний). Важливо зауважити, що SDA і SCL не є абсолютно відкритими; обидві панелі обладнані діодами захисту від електростатичного розряду для VDDIO та GND. Так як пристрої не виконують тактового розтягування, структура SCL призначена для використання як вхід з високим ступенем імпедансу, із заборонаю стоку.

Адреса пристрою складається з 7 розрядів і має вигляд 111011x. Перші 6 старших бітів (MSB) є незмінними. Останній біт залежить від значення SDO і може змінюватися під час роботи. Підключення SDO до GND призводить до підлеглої адреси 1110110 (0x76), а підключення його до VDDIO призводить до адреси 1110111 (0x77), яка збігається з адресою I2C BMP180. Важливо уникати залишання піна SDO плаваючим, оскільки це може призвести до невизначеної адреси I2C.

Інтерфейс I2C використовує такі контакти:

- SCK: послідовний годинник (SCL)
- SDI: дані (SDA)
- SDO: молодший біт адреси підлеглого пристрою (GND = «0», VDDIO = «1»)



Рисунок 2.6 – Багатобайтове читання I2C

SPI-інтерфейс підтримує режими SPI «00» (CPOL = CPHA = «0») та «11» (CPOL = CPHA = «1»). Автоматичний вибір між цими режимами визначається значенням SCK після спаду CSB. Існують два режими SPI: 4-провідний і 3-провідний, проте протокол залишається однаковим для обох. Вибір 3-провідного режиму здійснюється через встановлення «1» в регістрі spi3w_en. У 3-провідному режимі панель SDI використовується як панель даних.

Інтерфейс SPI використовує наступні з'єднання:

- CSB: вибір мікросхеми, активний низький рівень
- SCK: послідовний годинник
- SDI: послідовне введення даних; введення/виведення даних у 3-провідному режимі
- SDO: послідовний вихід даних; стан "висока імпедансність" (hi-Z) у 3-провідному режимі

CSB активується при низькому рівні і обладнаний вбудованим підтягуючим резистором. Дані на SDI зафіксовані пристроєм на передньому фронті SCK, тоді як SDO змінюються на спадному фронті SCK. Запуск комунікації відбувається при переході CSB в низький рівень і завершується при його поверненні в високий рівень; протягом цих переходів SCK на CSB повинен залишатися стабільним. Порядок взаємодії в рамках протоколу SPI представлений на рисунку 2.7.

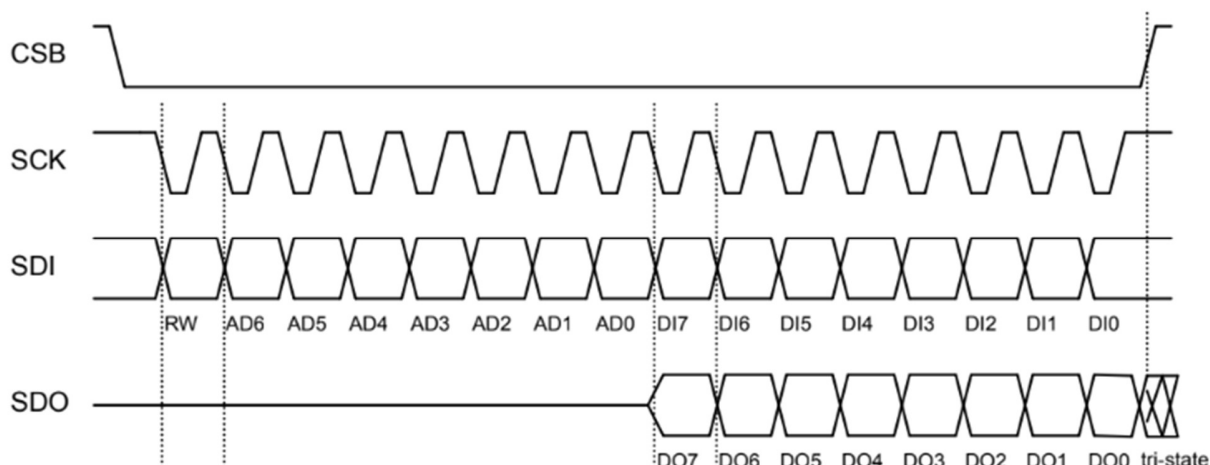


Рисунок 2.7 – Протокол SPI

У режимі SPI використовуються тільки 7 біт адрес регістра; MSB адреси регістра не використовується і замінюється бітом читання/запису ($RW = \langle 0 \rangle$ для запису та $RW = \langle 1 \rangle$ для читання). Приклад: доступ до адреси $0xF7$ здійснюється за допомогою адреси регістра SPI $0x77$. Для доступу на запис передається байт $0x77$, для доступу на читання передається байт $0xF7$.

Запис виконується шляхом активації низького рівня CSB та передачі пар контрольних байтів та даних регістрів. Керуючі байти включають адресу регістра SPI (повна адреса регістра без біта 7) та команду запису (біт 7 = $RW = \langle 0 \rangle$). Без підняття CSB можна здійснювати запис кількох пар. Транзакція завершується підняттям CSB. Інструкція щодо протоколу запису SPI наведена на рис. 2.8.

		Control byte								Data byte										Control byte								Data byte									
Start	RW	Register address (F4h)								Data register - address F4h								RW		Register address (F5h)								Data register - adress F5h								Stop	
CSB = 0	0	1	1	1	0	1	0	0		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	0		1	1	1	0	1	0	1		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	CSB = 1	

Рисунок 2.8 – Багатобайтний запис SPI

Читання ініціюється зниженням CSB та відправленням першого керуючого байта. Керуючі байти складаються з адреси регістра SPI (повна адреса регістра без біта 7) та команди читання (біт 7 = $RW = \langle 1 \rangle$). Після передачі керуючого байта

дані подаються з виводу SDO (або SDI у 3-провідному режимі); адреса регістра автоматично збільшується. Протокол читання SPI проілюстровано на Рис. 2.9.[41]

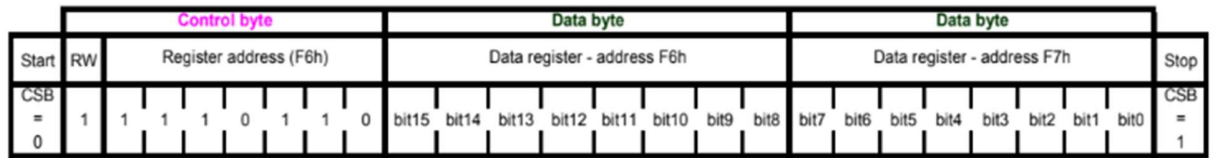


Рисунок 2.9 Багатобайтове читання SPI

2.3 Вибір мови програмування для мікроконтролера WeMos D1 mini

2.3.1 MicroPython

Одним із популярніших мов програмування вважається Python. Для написання кода, який би розумів мікроконтролер використовують MicroPython. MicroPython вміщує у себе компілятор та середовище виконання, які працюють безпосередньо на мікроконтролері. Для взаємодії з користувачем доступна інтерактивна консоль, що дозволяє безпосередньо виконувати команди. Крім того, MicroPython містить модулі, які дозволяють програмістам використовувати низькорівневі можливості мікроконтролера.

MicroPython володіє багатьма корисними функціями, включаючи інтерактивне запитання (REPL), і має достатньо компактний розмір, щоб вміститися в межах 512 КБ пам'яті та працювати з 16 КБ оперативної пам'яті.

Завдяки відкритій платформі, програмне забезпечення з MicroPython можна використовувати на численних популярних мікроконтролерах. Окрім PyBoard, платформи від розробника MicroPython можна використовувати на платах сімейства ESP, таких як ESP8266, ESP32, NodeMCU, WeMos D1, Espruino Pico, а також на Raspberry Pi Pico, BBC Micro: Bit, WiPy, STM32, ST Nucleo.

Сьогодні доступний різноманітний набір програмних інструментів для роботи з MicroPython, включаючи популярні редактори коду та інтегровані

середовища розробки (IDE). У MicroPython ESP8266 код для управління датчиками та моторами можна написати буквально за кілька рядків. Завдяки широкому вибору бібліотек, взаємодія з найпоширенішими компонентами стає дуже простою.

Одним із таких середовищ є uPyCraft IDE. Інтерфейс програми зображень на рис.2.10. Це інтегроване середовище розробки було спеціально створено для використання з MicroPython. Воно надає засоби для завантаження коду на плату та прошивки мікропрограми MicroPython. Останній інструмент є особливо корисним, оскільки він пропонує інтуїтивно зрозумілий, простий та швидкий спосіб запису прошивки MicroPython на контролер. uPyCraft IDE містить ліву бічну панель, яка відображає файли, що зберігаються на дошці, та файли, що містяться у вашій робочій області. Панель справа містить всі необхідні інструменти для встановлення зв'язку з дошкою, запуску та завантаження коду. Встановлення зв'язку із платою досить просте, якщо правильно обрати COM-порт. Зазвичай з'єднання встановлюється з першої спроби, і кнопка STOP ефективно припиняє виконання коду на платі, дозволяючи встановити нове з'єднання. Також є кнопка "Завантажити та запустити", яка, як із назви випливає, автоматично запускає код після завантаження. У порівнянні з іншими IDE, де потрібно вручну скидати плату для запуску коду, це значно спрощує процес.

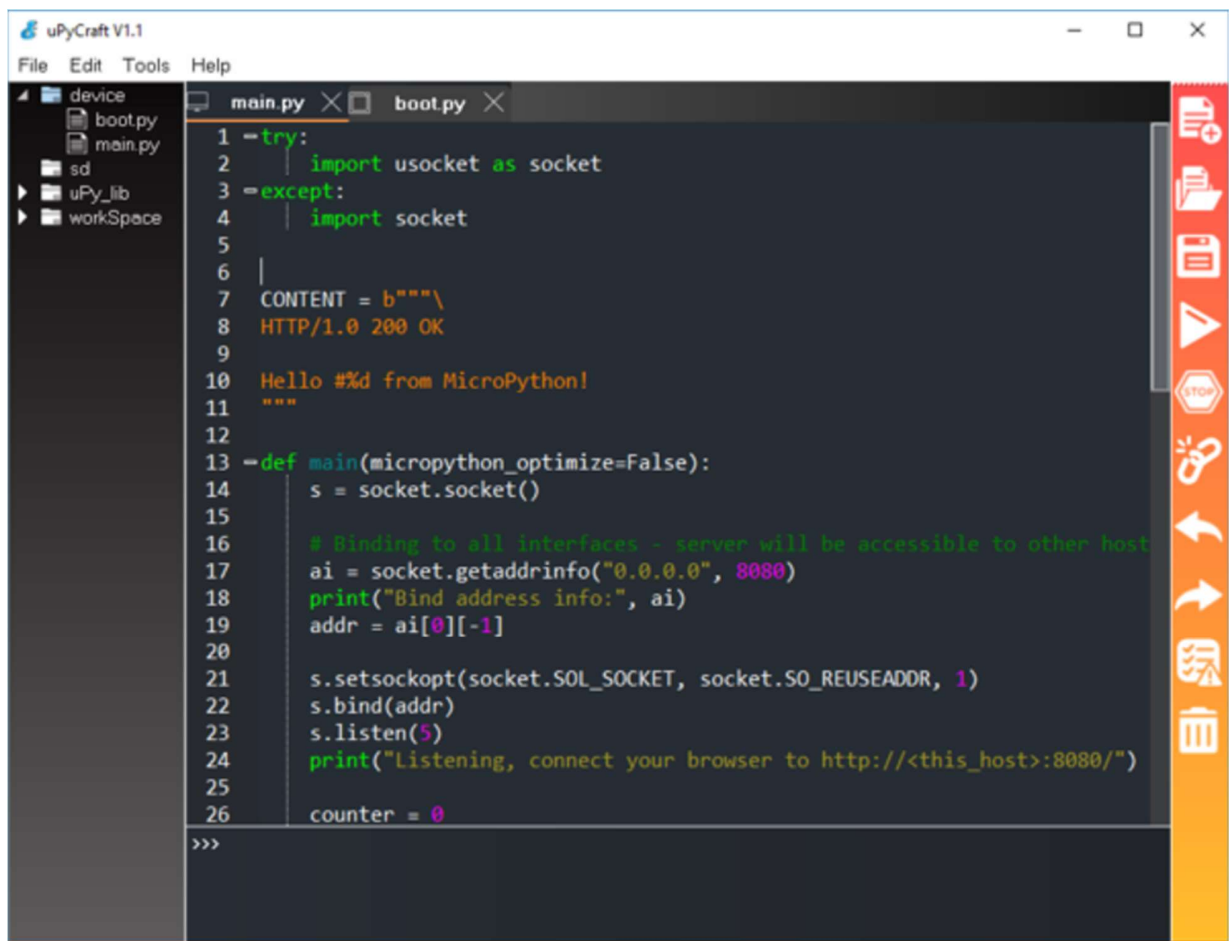


Рисунок 2.10 – Інтерфейс програми uPyCraft IDE

Є кілька недоліків у цій IDE: вона виводить всю інформацію про налагодження у REPL, що може призвести до плутанини серед початківців. Деякі інші IDE мають окреме вікно або файл, де виводиться інформація про налагодження. Іноді, якщо IDE не може встановити зв'язок з першої спроби, потрібно знову прошивати мікропрограму MicroPython на дошці. Це може бути досить непорозумілим, особливо при тестуванні та налагодженні коду. Крім того, існують проблеми із встановленням uPyCraft IDE, і після встановлення вона може працювати некоректно [42].

2.3.2 Arduino IDE

Найпопулярнішою мовою програмування для мікроконтролерів Arduino або сімейств плат ESP, WeMos D1 mini входить до цього сімейства, є Arduino IDE. Програма сумісна з операційними системами Windows, Linux та macOS, розроблена на Cі та C++. У неї включено весь необхідний інструментарій для зручної розробки власних проектів. Arduino IDE дозволяє написати код, оптимізувати роботу плати, отримувати дані про ефективність коду, встановлювати сторонні бібліотеки та створювати власні.

Якщо немає можливості встановити Arduino IDE, можна використовувати браузер для програмування плати в хмарному середовищі розробки. У цьому середовищі доступні функції роботи з кодом, відправлення написаного коду на плату, взаємодія з бібліотеками та моніторинг.

У розробці для Arduino, програми часто відомі як скетчі. Скетчі складаються в текстовому редакторі і зберігаються в файлах з розширенням .іо. У вбудованому текстовому редакторі можна копіювати, вставляти, шукати та замінити текст. Область повідомлень у вікні програми дозволяє користувачеві отримувати зворотній зв'язок, повідомляючи про події, включаючи помилки під час запису або експорту коду. Консоль може відображати текстовий вивід середовища Arduino, який містить повідомлення про помилки та інші дані. У правому нижньому куті вікна програми відображається модель плати, яка зараз використовується, а також послідовний порт, до якого вона підключена. Створення, відкриття, збереження та прошивка програм на пристрій можна виконати за допомогою кнопок на панелі інструментів. У програмі Serial Monitor використовується окрема кнопка. На рисунку 2.11 зображено інтерфейс Arduino IDE.

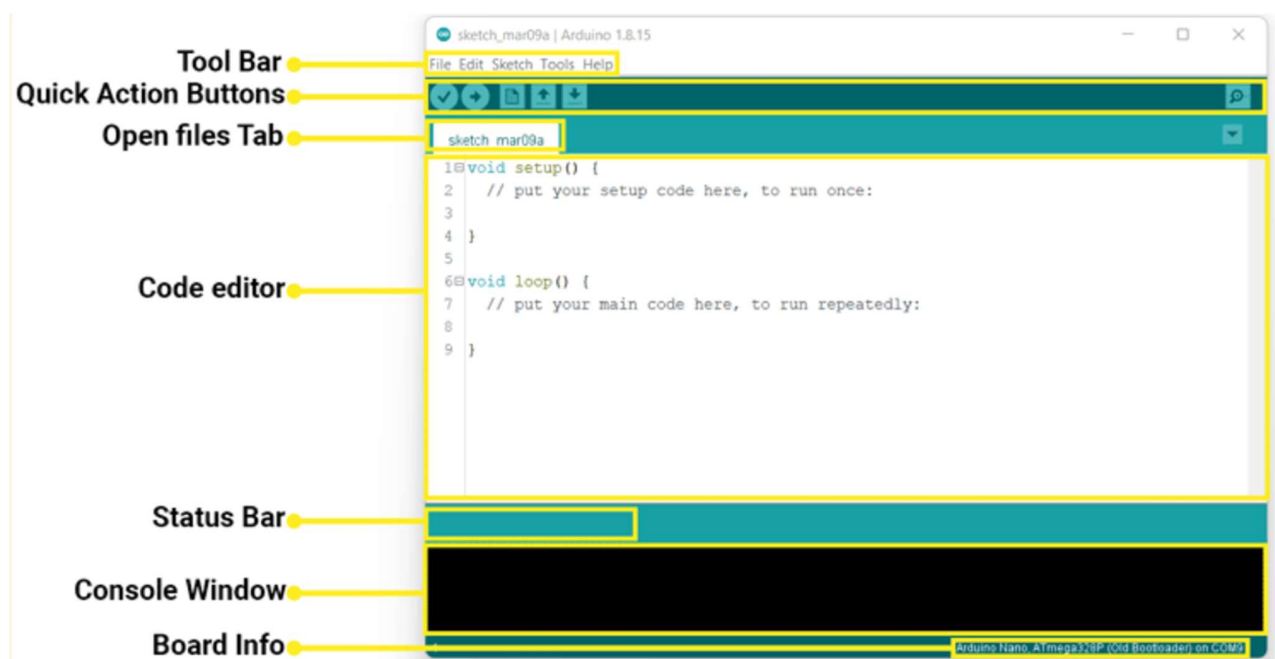


Рисунок 2.11 – Інтерфейс програми Arduino IDE

У процесі розробки для Arduino використовується концепція скетчбука: всі ваші програми (або скетчі) зберігаються в одному місці. Для перегляду їх потрібно вибрати меню **File > Sketchbook** або натискати кнопку **Open** на панелі інструментів. Папка для зберігання програм автоматично створюється при першому запуску середовища Arduino. Її місцезнаходження завжди можна змінити у вікні налаштувань програми. Середовище Arduino дозволяє працювати з програмами, які складаються з кількох файлів, кожен з яких відкривається у своїй окремій вкладці. Наприклад, це можуть бути файли Arduino (без розширення), С-файли (з розширенням `.c`), С++-файли (з розширенням `.cpp`) або файли заголовків (`.h`).

Перед тим як завантажити програму на контролер, необхідно налаштувати правильну плату та послідовний порт у меню **Tools > Board** та **Tools > Serial Port** відповідно. Нижче перелічені різновиди плат Arduino. На Mac-системах послідовний порт буде виглядати приблизно так: `/dev/tty.usbmodem241` (для Arduino Uno, Mega2560 або Leonardo) або `/dev/tty.usbserial-1B1` (для Duemilanove або старіших версій Arduino з USB), або `/dev/tty.USA19QW1b1P1.1` (для USB-UART перетворювачів Keyspan). На Windows-системах, можливий номер порту - COM1

або COM2 (для пристроїв з послідовним інтерфейсом), або COM4, COM5, COM6 і вище (для Arduino з USB) - визначити номер порту можна у Диспетчері пристроїв, знаходячи рядок "USB serial device" у розділі Порти. На Linux-системах послідовний порт буде виглядати як /dev/ttyUSB0, /dev/ttyUSB1 та інше.

Після вибору порту та плати, необхідно натискати кнопку Upload на панелі інструментів або вибрати пункт Upload з меню File. Після цього відбудеться скидання Arduino і розпочнеться процес завантаження програми на пам'ять контролера. У старих моделях (до Arduino Diecimila) функція авто-скидання відсутня, тому перед прошивкою таких пристроїв необхідно вручну натискати кнопку скидання на платі. Під час завантаження багатьох моделей Arduino світитимуть світлодіоди RX та TX. Після завершення процесу прошивки програма виведе відповідне повідомлення або помилку.

Завантаження програми в Arduino виконується за допомогою завантажувача - невеликої програми, що прошиває в пам'ять мікроконтролера. Завантажувач дозволяє завантажувати код без зовнішніх апаратних засобів. Завантажувач активується кілька секунд після скидання пристрою та запускає останній завантажений скетч. Під час активації завантажувача вбудований світлодіод, підключений до 13 ніжки контролера, буде мигати.

Бібліотеки розширюють можливості програм та включають додаткові функції, такі як робота з апаратними ресурсами, обробка даних і інші. Для включення бібліотеки слід обрати її у меню Sketch > Import Library. Після цього бібліотека буде скомпільована, а в початок програми буде додано один або кілька операторів #include. Важливо пам'ятати, що бібліотеки завантажуються на контролер разом із скетчем, тому кожна підключена бібліотека використовує додатковий обсяг пам'яті мікроконтролера. У випадку, якщо певна бібліотека не використовується у вашій програмі, можна видалити оператор #include з програмного коду.

Serial Monitor відображає отримані дані від Arduino на комп'ютері через послідовний інтерфейс, підтримуючи як USB-, так і звичайні версії Arduino. Щоб відправити дані зовнішньому пристрою, достатньо ввести текст у вікні програми та натиснути кнопку "Надіслати" або натискати клавішу Enter. У випадаючому списку потрібно вибрати швидкість передачі даних, що відповідає тій, яку ви вказали у функції `Serial.begin()` у своєму скетчі. Важливо зауважити, що на Mac і Linux Arduino буде скидатися при кожному підключенні програми до пристрою, що призводить до повторного виконання скетчу.

Також можна взаємодіяти з Arduino через інші програми, такі як Processing, Flash, MaxMSP і інші [43].

2.3.3 Scratch for Arduino

Проект від Массачусетського технологічного інституту S4A є варіантом вдосконалення мови візуального програмування для Arduino. За його допомогою можна скласти блоки коду, які автоматично конвертуються у готову прошивку для відповідних плат.

Цей додаток базується на MIT's Scratch і призначений для навчання програмуванню Arduino, спрямований як на дітей, так і на початківців. Основна ідея полягає в тому, щоб забезпечити більш інтуїтивне розуміння програмування через використання наочної мови. Навіть досвідчені програмісти Arduino можуть знайти цей додаток цікавим. Він становить відмінний вихідний пункт для тих, хто тільки починає програмувати на Arduino. Інтерфейс програми представлений на рис. 2.12.

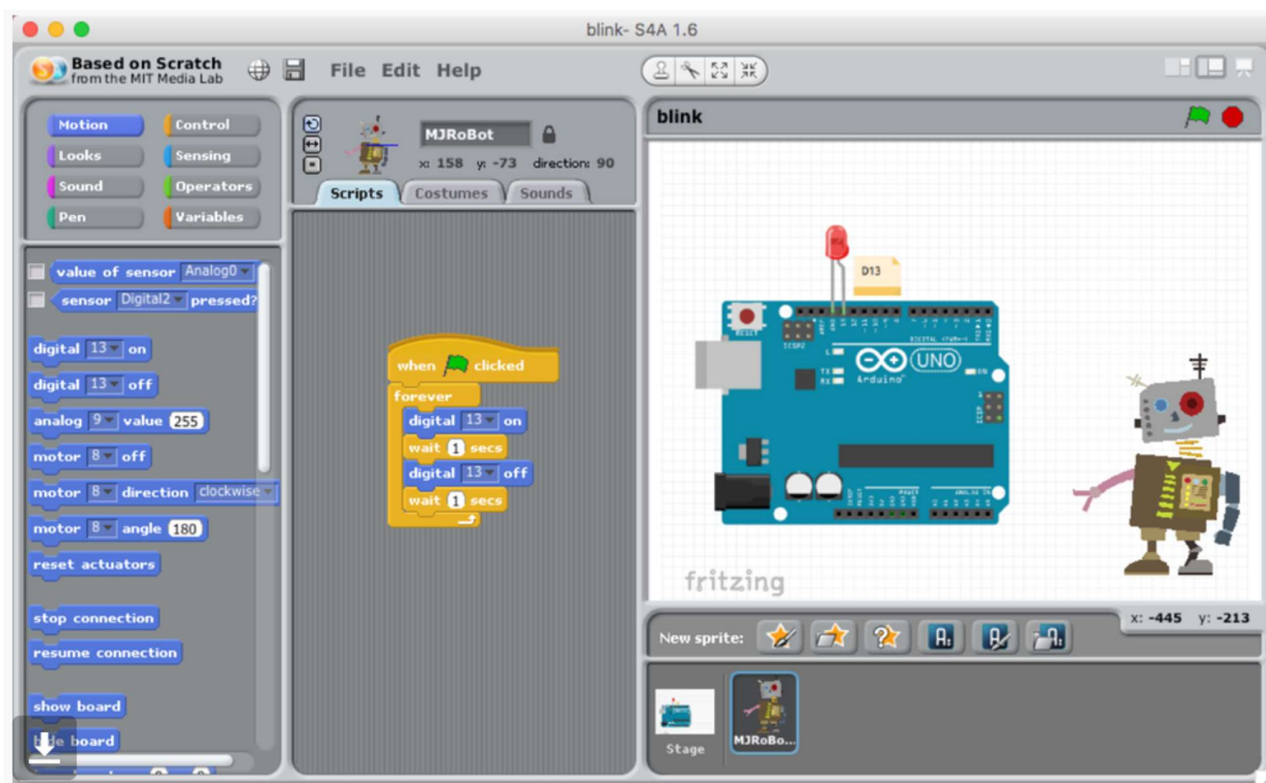


Рисунок 2.12 – Інтерфейс програми S4A (Scratch for Arduino)

Основні мікроконтролери Arduino відображаються у вигляді функціональних блоків, які охоплюють аналоговий та цифровий введення та виведення, а також більш високорівневі функції. Тут ви можете знайти блоки для управління стандартним та безперервним обертанням серводвигуна.

У S4A Arduino-плата виступає унікальним спрайтом. Спрайт Arduino автоматично визначає USB-порт, до якого підключена плата. Також можливо підключити кілька плат одночасно, просто додайте новий спрайт Arduino [44].

Для виконання моєї мети підходить програмне середовище Arduino IDE, оскільки вона легко встановлюється, легка в роботі і досить функціональна. Як вже зазначалось, є програмне середовище розробки, призначене для використання з платами Arduino та іншими, що підтримують мови програмування C/C++. У свою чергу, C++ — це вже мова із підтримкою, об'єктно-орієнтованої парадигми В основі цієї мови лежить мова C. Такий підхід дозволяє полегшити роботу програміста.

Отже, логіка та основа вебсервера буде написана на мові C/C++ із використанням сторонніх бібліотек.

У той же час необхідно одразу визначити, створивши веб-сайт за допомогою Arduino IDE, який використовує HTML/CSS для розмітки та стилізації, а JavaScript для логіки сторінки. Ці мови мають велику підтримку від своїх розробників і набули широкої популярності.

Мова гіпертекстової розмітки сторінок, відома як HTML, використовується для створення «скелету» веб-сайтів. У процесі перетворення мови в машинний код браузер виконує кожен рядок коду послідовно. Ви можете знайти відповідну інструкцію на офіційному веб-сайті розробника.

Мова каскадних таблиць стилів, відома як CSS, використовує основну розмітку та визначає стилі, такі як відступи, позиціонування елементів порівняно один з одним, кольори, шрифти тощо. Також можна знайти відповідну інструкцію на офіційному веб-сайті розробника.

JavaScript підтримує кілька парадигм програмування, включаючи імперативну, функціональну та об'єктно-орієнтовану. Він обробляється браузером і часто використовується для впровадження логіки веб-сторінок, надаючи сайту інтерактивність. Відповідні інструкції можна знайти на офіційному веб-сайті розробника [18].

Одночасно важливо обрати розумне програмне забезпечення для комп'ютерної розробки. Visual Studio є зручною програмою для розробки на Windows, яка містить багато технологій для написання програм на різних мовах програмування та платформах. Основні функції цієї програми є безкоштовними, а те, що вона розроблена Microsoft, не створює проблем для програмістів, які розробляють програми для комп'ютерів з операційною системою Windows. У цьому випадку можна використовувати технологію Windows Form з підтримкою мови XAML для розмітки сторінок і логіку додатку в C#.

XAML, також відомий як Extensible Application Markup Language, є мовою розширеної розмітки додатків, яка має синтаксис, схожий на HTML і CSS. Ця мова легко зрозуміла навіть для тих, хто раніше не працював з XML. Усю необхідну інформацію можна знайти на офіційному веб-сайті розробника [19].

Висновки до розділу 2

У даному розділі були відібрані основні компоненти, з яких буде зібрана метеостанція. Ключовими елементами є мікроконтролер WeMos D1 mini, модуль BMP280 та програмне середовище Arduino IDE.

Вибір цих компонентів був здійснений з огляду на їхню функціональність та сумісність між собою. Мікроконтролер WeMos D1 mini відзначається високою продуктивністю та можливістю ефективно взаємодіяти з іншими елементами системи. Модуль BMP280, який включає в себе датчик тиску, температури та вологості, обрано через його точність та надійність вимірювань.

Програмне середовище Arduino IDE обрано для розробки програмного забезпечення, оскільки воно відоме своєю зручністю використання та широким спектром бібліотек для роботи з мікроконтролерами. Цей вибір дозволяє ефективно реалізувати функціонал метеостанції та забезпечити стабільну та надійну роботу пристрою.

3 ПРОЕКТУВАННЯ ПРИЛАДУ І НАПИСАННЯ ПРОГРАМНОГО КОДУ WEMOS D1 MINI

3.1 Підключення датчика BME280 до плати WeMos D1 mini

Підключення є досить простим процесом. Спочатку вивід VIN модуля повинен бути з'єднаний з виводом 3.3V на WeMos D1 mini, а також з'єднуються виводи GND на обох платах. Потім вивід SCL повинен бути з'єднаний з тактовим виводом I2C D1 на платі WeMos D1 mini, і вивід SDA повинен бути з'єднаний з виводом даних I2C D2 на WeMos D1 mini.

На схемі (рис. 3.1) показано спосіб підключення модуля BME280 до плати WeMos D1 mini.

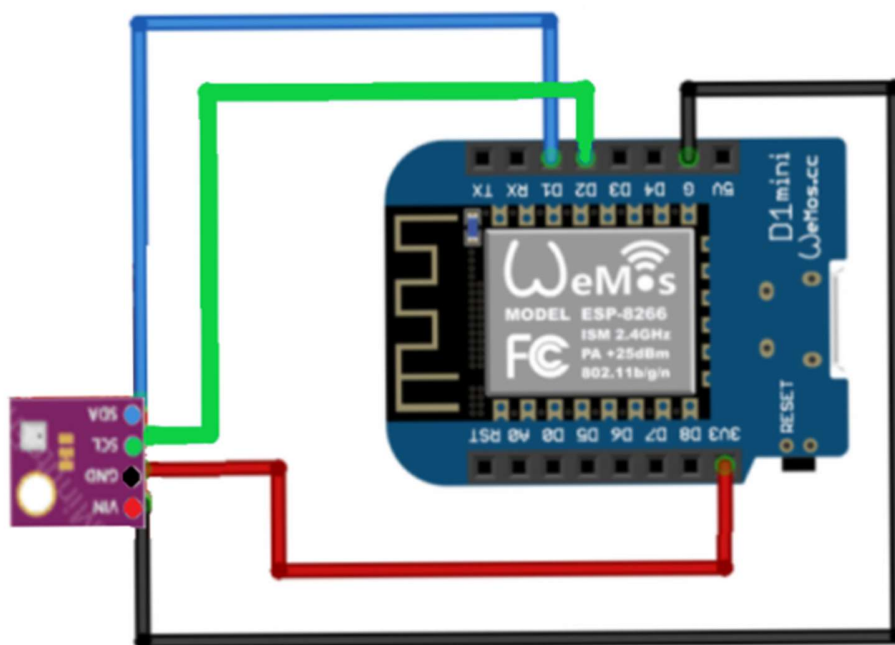


Рисунок 3.1 – Схема підключення датчика температури, вологості та тиску BME280 до WeMos D1 mini

3.2 Підготовка IDE Arduino

Для коректної роботи з платами, що використовують esp8266, слід встановити відповідне ядро та драйвер. Спочатку необхідно оновити менеджер плат, використовуючи користувацький URL. Відкрийте Arduino IDE, оберіть "Файл" → "Налаштування"(рис.3.2) . Після цього треба вставити наведений нижче URL в текстове поле " URL Менеджерів додаткових плат", яке розташоване у нижній частині вікна:

http://arduino.esp8266.com/stable/package_esp8266com_index.json.

На рисунку 3.3 зображено додавання плати ESP8266 до Arduino IDE через JSON URL.

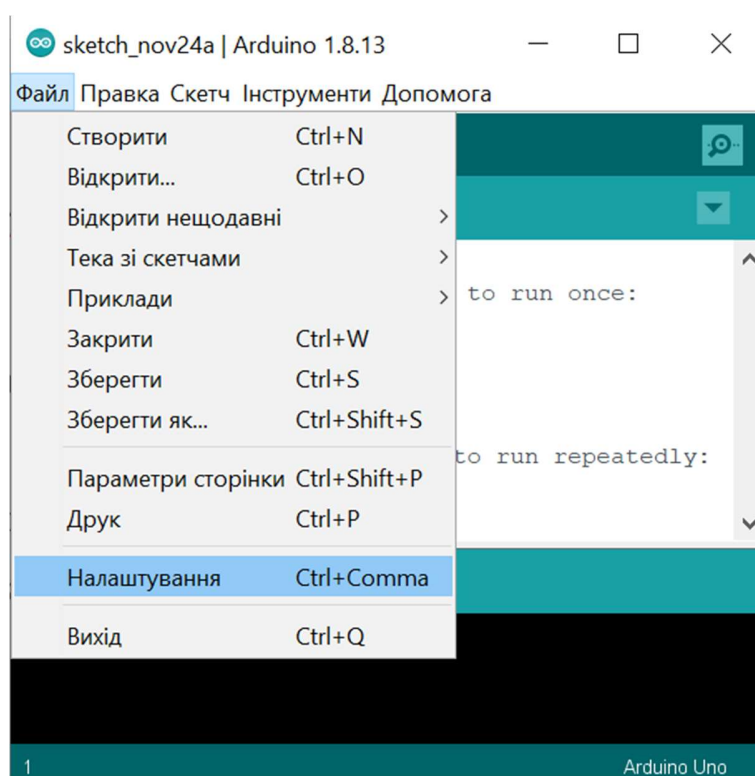


Рисунок 3.2 – Перехід в налаштування Arduino IDE

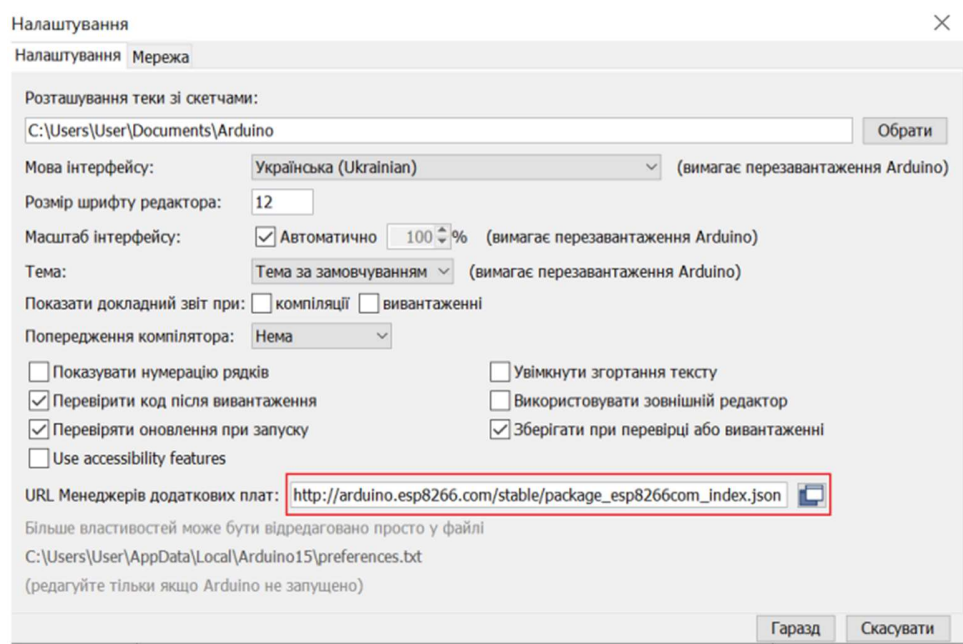


Рисунок 3.3 – Додавання плати ESP8266 до Arduino IDE через json URL

Потім перейдіть до Менеджера плат, виберіть "Інструменти" → "Плати" → "Менеджер плат". Застосуйте фільтр пошуку, введіть "esp8266". Оберіть відповідний запис і виконайте встановлення.

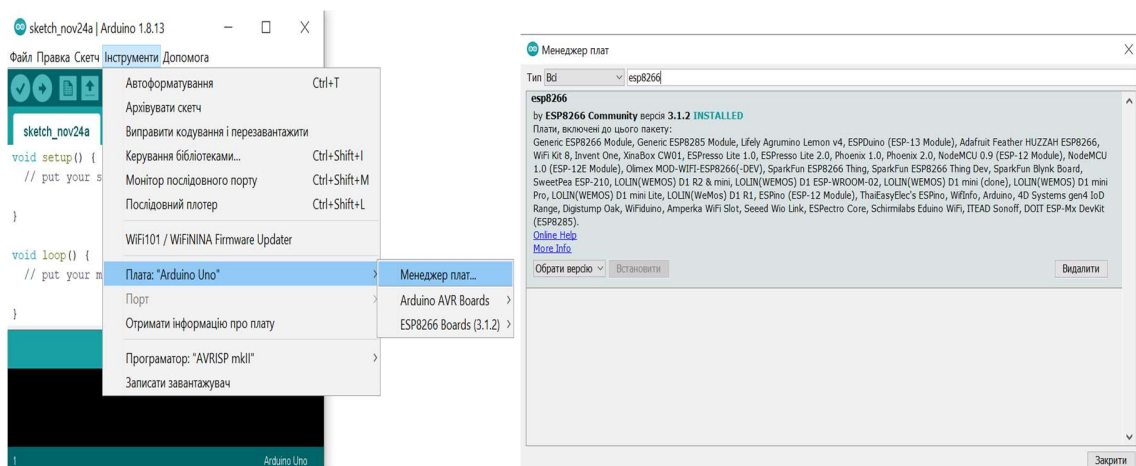


Рисунок 3.4 – Додавання ядра ESP8266 в менеджер плат Arduino IDE

Визначення та інструменти для плати ESP8266 включають в себе цілком новий набір компіляторів, таких як gcc, g++, та інші великі скомпільовані бінарні файли. Після успішного завершення встановлення з'явиться позначка "INSTALLED" біля відповідного запису.

3.3 Установка бібліотеки для сенсора BME280

Для взаємодії з модулем BME280 потрібно виконати ряд дій. На щастя, для спрощення цього процесу була створена бібліотека Adafruit BME280, що дозволяє легко отримувати дані про температуру, відносну вологість та атмосферний тиск за допомогою простих команд.

Щоб встановити бібліотеку, треба перейти до меню Скетч → Додати бібліотеку → Керування бібліотеками (рис.3.5). Зачекати, доки менеджер бібліотек завантажить індекс бібліотек та оновить список встановлених бібліотек.

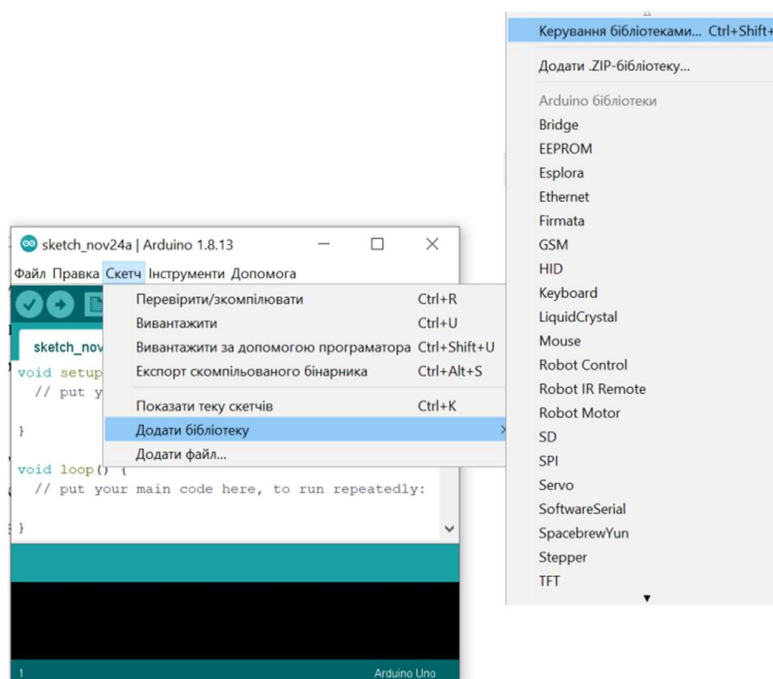


Рисунок 3.5 – Додавання бібліотек

Тепер необхідно здійснити фільтрацію результатів пошуку, введенням bme280. Там буде кілька записів, серед яких потрібно вибрати бібліотеку Adafruit BME280 від Adafruit. Треба вибрати цей запис і завантажити (рис.3.6).

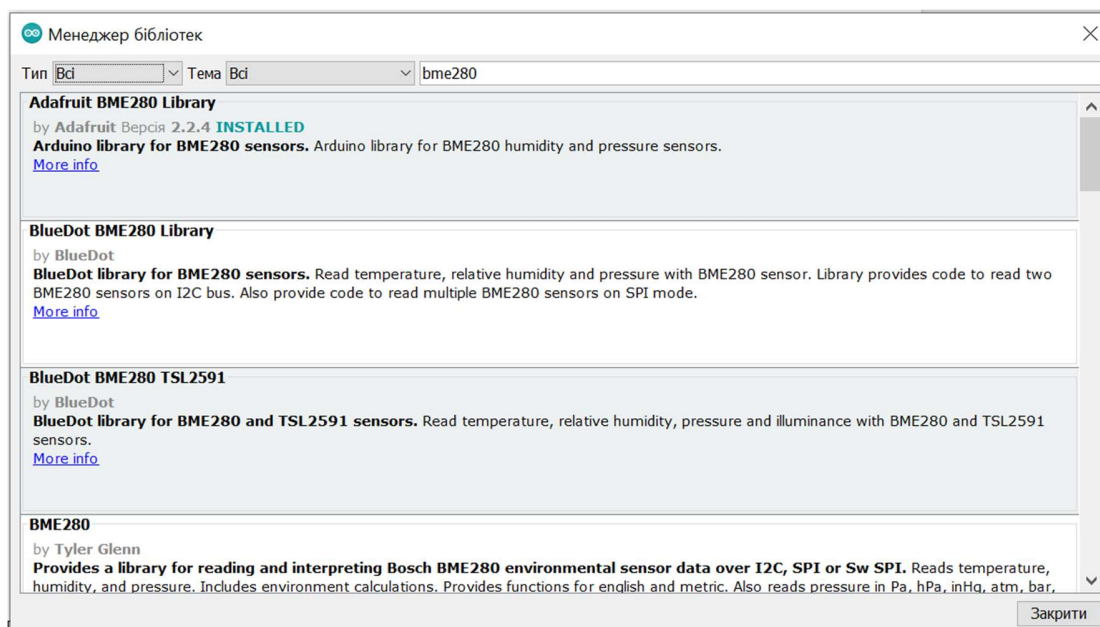


Рисунок 3.6 – Встановлення бібліотеки BME280 у середовищі Arduino IDE

Датчикова бібліотека BME280 використовує функціональний задній план Adafruit Sensor. Таким чином, також слід встановити його через менеджер бібліотек. На рисунку 3.7 зображено, як його можна знайти.

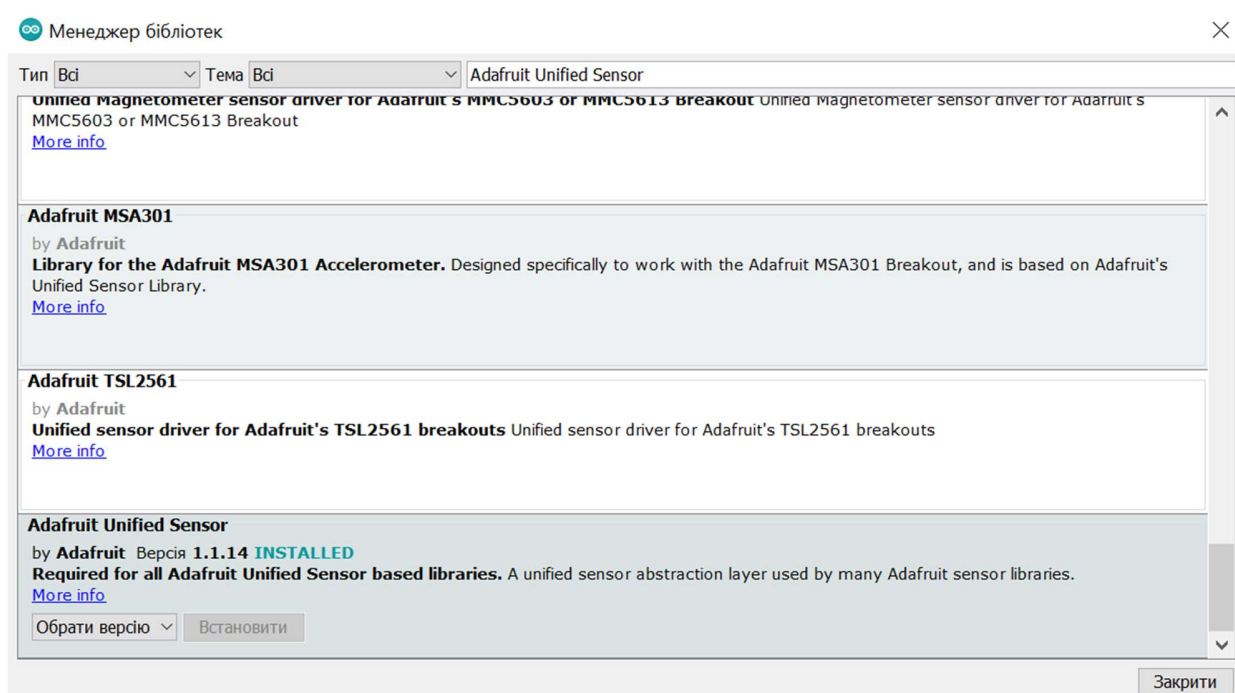


Рисунок 3.7 – Встановлення бібліотеки Adafruit Unified Sensor

3.4 Опис макету

У дипломному проекті представлені перші версії пристрою та відповідного програмного забезпечення, що створює можливість для його подальшого вдосконалення. Розділи розробки містять рішення, які функціонують, але можуть мати деякі недоліки. Таким чином, пристрій може бути представлений як макет, що надає можливість перевірити роботу як фізичної, так і віртуальної частини.

Під час розробки макету важливо визначитися з матеріалами та розмірами. Оскільки на пристрій подається напруга, матеріал корпусу повинен бути діелектриком для уникнення небезпечних ситуацій. У якості такого матеріалу було обрано корпус від світлодіодної лампи. Енергію пристрій отримує від USB-порту, який може бути підключений до power bank або Wi-Fi-роутера з USB-виходом. Це надає можливість віддалено перезавантажити метеостанцію разом з роутером за необхідності.

Для з'єднання модулів між собою використовуються дроти, і в даному випадку можна використовувати звичайні дроти для плат Arduino. В результаті отримується макет, який зображений на рисунку 3.8.



Рисунок 3.8 – Прототип пристрою

3.5 Створення програмного продукту

Для того щоб відображати дані про температуру, вологість, тиск та висоту на веб-сервері, необхідно сконфігурувати WeMos D1 mini у режимі станції (STA) та створити веб-сервер, який надаватиме веб-сторінки для будь-якого підключеного клієнта в існуючій мережі. У режимі STA WeMos D1 mini приєднується до наявної бездротової мережі WiFi, створеної вашим бездротовим маршрутизатором. Під час цього з'єднання мікроконтролер отримує IP-адресу від бездротового маршрутизатора, до якого він підключений. За допомогою цієї IP-адреси він може налаштовувати веб-сервер та надавати веб-сторінки для всіх пристроїв, які знаходяться в існуючій мережі Wi-Fi. Цей процес дозволяє зручно відслідковувати та керувати пристроєм через веб-інтерфейс з будь-якого підключеного пристрою в мережі.

Було створено версія коду, яка в подальшому вдосконалювалася. Планується подальший аналіз коду поетапно для кожної його частини. В ході цього процесу буде ретельно розглядатися та вдосконалюватися кожен блок програмного коду.

Перша версія коду наведена в додатку Б. Далі приведено розгорнуте роз'яснення програмного коду.

Як ми можемо бобачити з рис.3.9 програма (надалі Скетч) розпочинається з активації наступних бібліотек:

- Бібліотека ESP8266WebServer.h забезпечує спеціальні методи для взаємодії з Wi-Fi на платформі ESP8266 NodeMCU. Ці методи викликаються для здійснення підключення до бездротової мережі. Крім того, вона має декілька функцій, які спрощують налаштування сервера та обробку вхідних HTTP-запитів, при цьому не потребуючи докладного вдавання в деталі низькорівневої реалізації;

- Бібліотека Wire.h взаємодіє з будь-яким пристроєм, сумісним із шиною I2C, а не лише з BME280;

- Бібліотеки Adafruit_BME280.h та Adafruit_Sensor.h є апаратно-залежними бібліотеками, які відповідають за обробку низькорівневих функцій.

```
#include <ESP8266WebServer.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

Рисунок 3.9 – Фрагмент коду 1

Далі створюється об'єкт датчика та змінні для зберігання значень температури, вологості, тиску та висоти. Ми це можемо побачити на рис.3.10.

```

,
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME280 bme;

float temperature, humidity, pressure, altitude;

```

Рисунок 3.10 – Фрагмент коду 2

Налаштовуючи ESP8266 в режимі станції (STA), ми забезпечуємо його автоматичне приєднання до існуючої мережі Wi-Fi. Це важливий крок, оскільки дозволяє пристрою взаємодіяти з іншими пристроями та використовувати доступ до Інтернету. Для здійснення цього з'єднання, ми передаємо ESP8266 ім'я (SSID) та пароль нашої Wi-Fi мережі.

Після успішного приєднання до мережі ми ініціюємо роботу веб-сервера, який буде слухати запити на порті 80. Це дозволяє ESP8266 обслуговувати HTTP-запити та взаємодіяти з підключеними пристроями через веб-інтерфейс. Створення веб-сервера на ESP8266 в режимі станції надає можливість керувати та відслідковувати дані пристрою в зручний спосіб через браузер. Цей фрагмент коду відображен на рис.3.11.

```

/* Введіть SSID та пароль від вашої мережі */
const char* ssid = "YourNetworkName";    // SSID
const char* password = "YourPassword";    // пароль

```

Рисунок 3.11 – Фрагмент коду 3

Вміст функції setup(). У рамках функції setup() ми проводимо налаштування нашого HTTP сервера перед його активацією. В межах функції setup() ми також проводимо початкову конфігурацію параметрів HTTP сервера. Визначається режим роботи та обирається порт для правильного оброблення та маршрутизації вхідних

HTTP-запитів. Зазначається основний шлях, який буде відповідати за обробку запитів та відображення відповідей на веб-сторінці.

Перш за все, відбувається ініціалізація послідовного зв'язку з ПК, що важливо для взаємодії та відладки. Далі, ми ініціалізуємо об'єкт BME, використовуючи функцію `begin()`. Ця функція забезпечує ініціалізацію інтерфейсу I2C із зазначеною адресою I2C (0x76), перевіряє вірність ідентифікатора чіпа та запускає програмне перезавантаження мікросхеми. Після цього вона очікує завершення процесу калібрування датчика після запуску. Фрагмент код з функцією `begin()` відображен на рис. 3.12.

Цей крок важливий для забезпечення коректного збору даних від датчика та готовності сервера до прийому та обробки HTTP-запитів. Спрограмоване перезавантаження та чекання на завершення калібрування забезпечують стабільність та надійність операцій датчика.

```
Serial.begin(115200);  
delay(100);  
  
bme.begin(0x76);
```

Рисунок 3.12 – Фрагмент коду 4

Після виклику функції `WiFi.begin()`, WeMos D1 mini намагається встановити з'єднання з вказаною Wi-Fi мережею, використовуючи передані параметри SSID та пароль. Цей етап є ключовим для нормального функціонування пристрою у мережі, оскільки успішне підключення є передумовою для взаємодії з іншими пристроями та доступу до Інтернету.

Проте важливо враховувати, що процес підключення може зайняти певний час, особливо при першому запуску або у випадку невірного введення інформації про мережу. Фрагмент коду с функцією `WiFi.begin()` відображена на рис.3.13.

```

Serial.println("Connecting to ");
Serial.println(ssid);

// підключитися до вашої локальної wi-fi мережі
WiFi.begin(ssid, password);

```

Рисунок 3.13 – Фрагмент коду 5

В процесі спроби підключення WeMos D1 mini до мережі важливо відстежувати стан цього підключення, щоб ефективно керувати подальшим виконанням програми. З цією метою ми використовуємо функцію `WiFi.status()` для періодичної перевірки статусу підключення.

Цей крок є необхідним для виявлення можливих проблем, таких як невірний пароль або недоступність обраної мережі. Якщо статус підключення підтверджує успішне з'єднання, програма може продовжити свою роботу. У випадку невдачі іде повторна спроба підключення. Цей фрагмент коду зображено на рис.3.14.

```

// перевірити, чи виконано підключення wi-fi мережі
while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.print(".");
}

```

Рисунок 3.14 – Фрагмент коду 6

Після успішного підключення WeMos D1 mini до бездротової мережі, програма виводить на монітор послідовного порту IP-адресу, призначеного самому WeMos D1 mini. Цю інформацію виводиться за допомогою методу `WiFi.localIP()`. Цей етап має важливе значення, оскільки IP-адреса є ключовою інформацією для подальшої взаємодії з пристроєм у мережі. Фрагмент коду з функцією `WiFi.localIP()` зображено на рис.3.15 .

```
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());
```

Рисунок 3.15 – Фрагмент коду 7

Для обробки вхідних HTTP-запитів необхідно визначити, який код виконувати при надходженні певного URL. Для цього використовується метод `on()`. Цей метод приймає два параметри: перший – це URL-шлях, а другий – ім'я функції, яку ми хочемо виконати під час входу до цього URL.

Наведений у документації код показує, що при отриманні HTTP-запиту до кореневого шляху (`/`), сервер запускає функцію `handle_OnConnect()`. Важливо відзначити, що вказана URL-адреса є відносним шляхом.

Далі визначається, як сервер повинен реагувати, якщо клієнт запитує будь-який URL, який не співпадає із зазначеними у `server.on()`. У такому випадку сервер повинен повернути відповідь із статусом HTTP 404 (Not Found, «сторінка не знайдена») та повідомленням для користувача. Ці дії виконуються в окремій функції, а метод `server.onNotFound()` вказує серверу, що робити при отриманні запиту на URI, який не був визначений за допомогою `server.on()`. Даний фрагмент коду представлений на рис. 3.16.

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

Рисунок 3.16 – Фрагмент коду 8

Для запуску сервера тепер достатньо викликати метод `begin()` на об'єкті `server`. Фрагмент коду відображено на рис.3.17.

```
server.begin();  
Serial.println("HTTP server started");
```

Рисунок 3.17 – Фрагмент коду 9

Вміст функції `loop()`. Для обробки реальних вхідних HTTP-запитів необхідно викликати метод `handleClient()` на об'єкті `server`. Крім цього, необхідно створити функцію, яку ми прикріпили до кореневого URL за допомогою `server.on()`.

На початку цієї функції проводиться отримання показань температури кожного датчика. Для відповіді на HTTP-запит використовується метод `send()`. Навіть якщо цей метод може бути викликаний з різними аргументами, його простіша форма включає код HTTP-відповіді, тип контенту і сам контент.

У нашому випадку ми відправляємо код 200 (один із кодів статусу HTTP), що відповідає "ОК". Далі ми вказуємо тип контенту як "text/html". Завершальним етапом є виклик функції користувача `SendHTML()`, яка генерує динамічну HTML-сторінку із показниками температури, вологості, тиску і висоти.

Цей підхід дозволяє відправляти користувачеві інформативну та структуровану відповідь на його запит, створюючи зручний інтерфейс для отримання даних в реальному часі. Аналогічно, необхідно створити функцію для обробки сторінки з помилкою 404. Цей фрагмент коду відображено на рис.3.18.

```

void loop()
{
    server.handleClient();
}

void handle_OnConnect()
{
    temperature = bme.readTemperature();
    humidity = bme.readHumidity();
    pressure = bme.readPressure() / 100.0F;
    altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
    server.send(200, "text/html", SendHTML(temperature, humidity, pressure, altitude));
}

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

```

Рисунок 3.18 – Фрагмент коду 10

Генерація HTML-сторінки для відображення в браузері. Функція `SendHTML()` відповідає за створення веб-сторінки кожен раз, коли веб-сервер ESP8266 отримує запит від веб-клієнта. Вона просто об'єднує HTML-код у великий рядок і повертає його до функції `server.send()`, яку ми обговорювали раніше. Ця функція приймає параметри для динамічного генерування HTML-контенту, таких як показники температури, вологості, тиску і висоти.

Основний текст, який завжди передує, – це оголошення `<!DOCTYPE>`, що вказує на те, що ми надсилаємо HTML-код. Далі елемент `<meta>`, який встановлює властивості `viewport`, робить веб-сторінку адаптивною у будь-якому веб-браузері. За цим слідує тег `<title>`, що встановлює заголовок сторінки. Цей фрагмент коду відображено на рис.3.19.

```

String SendHTML(float temperature, float humidity, float pressure, float altitude)
{
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
    ptr += "<title>ESP8266 Weather Station</title>\n";
}

```

Рисунок 3.18 – Фрагмент коду 11

Оформлення веб-сторінки. Далі поданий фрагмент CSS-коду призначений для оформлення зовнішнього вигляду веб-сторінки. Ми визначаємо шрифт Helvetica та вказуємо, що контент буде відображатися у вигляді inline-block, центруючи його. Далі наведено код, який визначає кольори, шрифти та відступи для елементів body, H1 та p. Цей фрагмент коду відображено на рис.3.20.

```
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";
```

Рисунок 3.20 – Фрагмент коду 12

Потім ми встановлюємо заголовок на веб-сторінці, який буде служити визначеною назвою для вмісту. Для динамічного відображення актуальних показників температури, вологості, тиску та висоти, ми вбудовуємо ці значення у відповідні теги параграфів. Крім того, для коректного відображення символу градуса використовується HTML-сутність °.

Важливо відзначити, що цей підхід дозволяє нам динамічно оновлювати дані на веб-сторінці і забезпечує користувача актуальною інформацією щодо погодних умов або інших вимірів, які ми моніторимо. Такий підхід дозволяє створювати інтерактивні та змістовні веб-інтерфейси для спостереження за параметрами пристрою. Цей фрагмент коду відображено на рис.3.21.

```

ptr += "<div id=\"webpage\">\n";
ptr += "<h1>ESP8266 Weather Station</h1>\n";
ptr += "<p>Temperature: ";
ptr += temperature;
ptr += "&deg;C</p>";
ptr += "<p>Humidity: ";
ptr += humidity;
ptr += "%</p>";
ptr += "<p>Pressure: ";
ptr += pressure;
ptr += "hPa</p>";
ptr += "<p>Altitude: ";
ptr += altitude;
ptr += "m</p>";
ptr += "</div>\n";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}

```

Рисунок 3.21 – Фрагмент коду 12

Після завантаження скетчу активуйте монітор послідовного порту із швидкістю 115 200 біт/с, а потім натискайте кнопку RST на WeMos D1 mini. Система виведе динамічну IP-адресу, отриману від вашого маршрутизатора, і повідомить, що сервер HTTP успішно запущений. Вам потрібно ввести цей IP-адресу у веб-браузері, який виберете, а згодом WeMos D1 mini відобразить сторінку з вимірами температури, вологості, тиску та висоти, отриманими від датчика BME280.

Спостереження за першою версією веб-сайту, яка показана на рис. 3.22, відбувалися у закритому приміщенні. Це дозволяє оцінити роботу системи та її здатність відображати актуальні дані з датчика в реальному часі.

ESP8266 Weather Station

Temperature: 25.91°C

Humidity: 45.32%

Pressure: 950.97hPa

Altitude: 531.91m

Рисунок 3.22 – Виведення значень BME280 на веб-сервері (перша версія)

Друга версія коду наведена в додатку В. В цій версії коду було вдосконалено, частину коду яка відповідає за візуальний ефект с HTML-сторінки. Програмісти, такі, як ми, часто відчують нестійкість стосовно дизайну, але важко перебільшити, як важливою є привабливість та професійний вигляд веб-сторінки. Маленькі вдосконалення можуть значно підняти її рівень. Я вирішив внести зміни в стиль нашої попередньої HTML-сторінки, щоб надати їй сучасний та привабливий вигляд. Однією з ключових трансформацій було вдосконалення функції `SendHTML()`, яка тепер пройшла значні зміни для поліпшення її зовнішнього вигляду та функціональності.

Порівнюючи цю функцію із попередньою, можна відзначити їхню схожість, за винятком кількох важливих змін. На нашій веб-сторінці ми призначили шрифт Open Sans від Google, щоб надати їй сучасний вигляд. Важливо врахувати, що для перегляду цього шрифту потрібне активне підключення до Інтернету на пристрої, який ви використовуєте для доступу до сторінки, оскільки шрифти Google завантажуються. На рисунку 3.23 зображено використання шрифту Open Sans у коді.

```
ptr += "<link href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600' rel='stylesheet'>";
```

Рисунок 3.23 – Використання шрифту Open Sans у коді

З метою поліпшення естетичного вигляду веб-сайту ми використовуємо зображення у вигляді іконок для відображення різних параметрів, таких як температура, тиск і вологість. Іконки, що представляють ці показники, використовують масштабовану векторну графіку (SVG), яка визначена у тезі <svg>. Важливо відзначити, що створення графіки у форматі SVG не вимагає особливих навичок програмування, і для цього можна використовувати будь-який редактор SVG. Наш вибір припав на іконки які відображені на рис.3.24 у форматі SVG, щоб забезпечити ефективне та привабливе відображення графіки сторінки.



Рисунок 3.24 – SVG іконки які використовуються в коді

Покрім візуального вдосконалення, ми додали функціональність для динамічного завантаження даних з датчиків, з метою уникнення необхідності вручну оновлювати сторінку при кожному запиті. Для цього ми використали асинхронний Javascript та XML (AJAX). Цей метод дозволяє асинхронно (у фоновому режимі) запитувати дані із сервера, не вимагаючи при цьому повного оновлення сторінки.

Для ефективної реалізації AJAX на веб-сторінках, за звичай використовується об'єкт XMLHttpRequest у мові JavaScript. Він виконує GET-запит на сервер та динамічно оновлює елементи на сторінці, забезпечуючи миттєвий обмін даними між користувачем та сервером. Необхідно підкреслити, що AJAX не є самостійною

новою технологією чи окремою мовою програмування; це лише новий підхід до використання існуючих технологій. Серед його можливостей варто відзначити:

- Здатність запитувати дані із сервера після завантаження сторінки, що дозволяє динамічно оновлювати вміст без повторного завантаження всієї сторінки.
- Можливість отримувати дані із сервера під час роботи сторінки, що робить можливим асинхронний обмін інформацією без порушення користувацького досвіду.
- Функціонал для відправки даних на сервер у фоновому режимі, що полегшує взаємодію з сервером без блокування основного інтерфейсу користувача.

Нижче поданий частина скрипта AJAX рис.3.25, який буде використовуватися для впровадження.

```
ptr +="<script>\n";
ptr +="setInterval(loadDoc,1000);\n";
ptr +="function loadDoc() {\n";
ptr +="var xhttp = new XMLHttpRequest();\n";
ptr +="xhttp.onreadystatechange = function() {\n";
ptr +="if (this.readyState == 4 && this.status == 200) {\n";
ptr +="document.body.innerHTML =this.responseText}\n";
ptr +="};\n";
ptr +="xhttp.open(\"GET\", \"/\", true);\n";
ptr +="xhttp.send();\n";
ptr +="}\n";
ptr +="</script>\n";
```

Рисунок 3.25 – Фрагмент код з використанням AJAX

Скрипт розпочинається з використання тега `<script>`, оскільки AJAX-скрипт фактично є JavaScript, тому його необхідно розміщувати в межах тегу `<script>`. Для повторного виклику цієї функції ми будемо використовувати функцію `setInterval()` мови JavaScript. Ця функція приймає два параметри: функцію для виконання та інтервал часу (у мілісекундах), з яким цю функцію слід викликати.

Центральним елементом цього скрипта є функція `loadDoc()`. Усередині цієї функції створюється об'єкт `XMLHttpRequest()`. Цей об'єкт використовується для здійснення запиту даних із веб-сервера.

Функція `xhttp.onreadystatechange()` викликається при зміні значення `readyState`. Властивість `readyState` зберігає стан `XMLHttpRequest`, який може мати значення від 0 до 4, представляючи різні етапи обробки запиту.

Властивість `status` містить статус об'єкта `XMLHttpRequest`, де код 200 вказує на успішний запит, а 403 чи 404 можуть свідчити про помилку чи відмову.

Коли `readyState` дорівнює 4, а `status` - 200, це означає, що відповідь від сервера готова. На цьому етапі оновлюється вміст елемента `body`, який містить значення температури, вологості, тиску та висоти. HTTP-запит ініціюється за допомогою функцій `open()` та `send()`. На рис.3.26 приведена скріншот показників с вдосконаленим кодом.

ESP8266 Weather Station

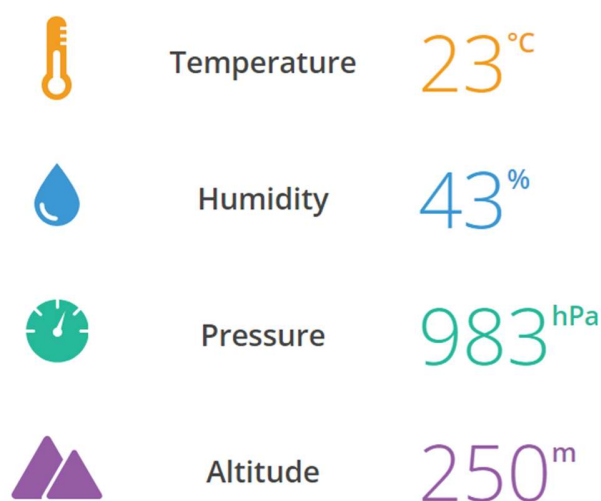


Рисунок 3.26 – Виведення значень BME280 на веб-сервері (друга версія)

Висновки до розділу 3

У цьому розділі було представлено схему підключення датчика BME280 до плати Wemos D1 mini, а також викладені кроки щодо встановлення ядра та

драйверів, необхідних для коректної роботи плати. Також було розглянуто необхідні налаштування бібліотеки для оптимального використання датчика BME280.

Додатково були створені та описані два варіанти програмного забезпечення, в яких Wemos D1 mini функціонує як пристрій управління. У цих програмах плата підключається до існуючої мережі Wi-Fi та створює веб-сервер. Коли інший пристрій підключається до цього веб-сервера через мережу Wi-Fi, ESP8266 зчитує дані про температуру, вологість, атмосферний тиск і висоту з датчика BME280, а потім передає ці дані до веб-браузера пристрою, відображаючи їх на веб-сторінці.

ВИСНОВКИ

У даній дипломній роботі вичерпно описано процес проектування макету автоматизованої метеостанції з використанням мікроконтролера для управління. Розглянуті різновиди датчиків та принципи їхньої роботи, де висвітлено особливості сучасних метеостанцій.

В роботі висвітлено етапи розробки функціональної схеми пристрою, а також обґрунтовано вибір елементної бази, що включає в себе мікроконтролер, датчики температури, вологості та тиску. Для реалізації макету обрана плата Wemos D1 Mini, яка використовує мікроконтролер ESP8266, а також цифровий датчик BME280 який видає показник вологості та температури, а також параметри тиску.

Детально розглянуто розробку керуючої програми та взаємодію з кожним з датчиків автоматизованої метеостанції. Програма була реалізована в середовищі Arduino IDE. Після зібрання всіх необхідних елементів, метеостанцію вдало складено в єдиний функціональний пристрій.

Щоб забезпечити захист від зовнішніх механічних пошкоджень, всі компоненти були вміщені в корпус. Це додатково гарантує надійність та довговічність метеостанції в експлуатації.

Були проведені дослідження у сфері систем автоматичного керування метеостанціями, в результаті чого було проаналізовано та обрано необхідні компоненти, спрямовані на досягнення оптимальної функціональності та високої надійності метеостанції. При виборі мови програмування дотримувалися вимог проекту, максимізуючи ефективність реалізації функцій вимірювань. Результатом став розроблений пристрій, який відповідає визначеним функціональним вимогам, забезпечуючи точні та стабільні вимірювання.

Була розроблена автоматизована система для керування процесом вимірювань, що робить метеостанцію ефективним інструментом для збору метеоданих. Також були виявлені можливості для подальшого розвитку та

удосконалення функціональності метеостанції, зокрема, розширення спектру вимірюваних параметрів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коваленко Ю. Л. Метеорологія і кліматологія: конспект лекцій / Ю. Л. Коваленко. – Х.: ХНУМГ ім. О. М. Бекетова, 2018. – 65 с.
2. Як винайдено термометр [Електронний ресурс] // Захоплива фізика. – 2004. – URL: <https://www.afizika.ru/zanimatelniestati/171-kakizobrelitermometr>.
3. Історія термометра [Електронний ресурс] // Музей історії розвитку Гідрометцентра Росії. – 2012. – Режим доступу до ресурсу: <http://museum.meteorf.ru/present/pdf/thermo.pdf>.
4. A brief history of thermometers [Електронний ресурс] // Guardian News & Media Limited. – 2003. – URL: <https://www.theguardian.com/science/2003/aug/06/weather.environment>.
5. Хромов С.П. Метеорологія і кліматологія: класичний університетський підручник / Петросянц М.А. : МГУ ім. М.В. Ломоносова, 2006. – 210 с.
6. Будова та види датчиків температури в кліматичних системах [Електронний ресурс] // DS Electronics. – 2005. – URL: <https://ds-electronics.com.ua/ua/support/blog/termoregulatory/ustrojstvo-i-vidydatchikov-temperature-v-klimaticheskikh-sistemah/>.
7. Авіаційні термометри [Електронний ресурс] // Кіровоградська Льотна Академія Національного Авіаційного Університету. – 2018. – URL: <https://studfile.net/preview/6810193/page:17/>.
8. Different Types of Thermometers [Електронний ресурс] // Instrumentation Tools. – 2021. – URL: <https://instrumentationtools.com/different-types-thermometers/>.
9. Руководство по метеорологическим приборам и методам наблюдений. // © Всемирная Метеорологическая Организация. – 2010. – С. I.2–12, 2.4.1, 72 с.
10. Датчик вологості та температури GY-21 HTU21 [Електронний ресурс] // arduino.ua – URL: <https://arduino.ua/prod1222-datchikvlajnosti-i-temperaturi>.

11. Призначення електронних термометрів [Електронний ресурс] // АО «ЭКСИС». – 2015. – URL:
<https://www.eksis.ru/materials/articles/elektronnye-termometry.php>.
12. Пірометри [Електронний ресурс] // Вінницький національний технічний університет. – 2015. – URL: <https://studfile.net/preview/3907292/page:5/>.
13. Інфрачервоний датчик температури [Електронний ресурс] // VirtualExpo. – 2021. – URL:
<https://www.directindustry.com.ru/prod/melexis/product-12118-210268.html>.
14. Мишанський, М. А. Вологість повітря і способи її вимірювання / М. А. Мишанський // Збірник доповідей Республіканської науково-технічної конференції аспірантів, магістрантів і студентів "Інновації в техніці і технології дорожньо-транспортного комплексу". Секція "Дорожна кліматологія". У 6 ч. Ч. 3 / наук. кер. І. І. Леонович. – Мінськ : БНТУ, 2013.
15. Про термометри: що це таке, принцип дії, підключення, застосування [Електронний ресурс] // Інформаційний інтернет-сайт "ASUTPP". – 2021. – URL:
<https://www.asutpp.ru/termopary.html>.
16. Винаходи Леонардо да Вінчі (частина 3) [Електронний ресурс] // Історія винаходів. – 2017. – URL:
<http://kakizobreli.ru/izobreteniya-leonardo-da-vinchi-chast-3/#gigrometr>.
17. Історія виникнення і розвитку Гігрометра [Електронний ресурс] // Бобродобро – URL: <https://prog.bobrodobro.ru/102141>.
18. Водяна пара, вологість повітря. [Електронний ресурс] // Гіпермаркет знань. – 2008. – URL:
https://edufuture.biz/index.php?title=38._%D0%92%D0%BE%D0%B4%D1%8F%D0%BD%D0%B0_%D0%BF%D0%B0%D1%80%D0%B0,_%D0%B2%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%81%D1%82%D1%8C_%D0%BF%D0%BE%D0%B2%D1%96%D1%82%D1%80%D1%8F.

19. Датчики температури. Види і принцип дії, Як вибрати [Електронний ресурс] // ЕЛЕКТРОСАМ.РУ. – 2021. – URL:

<https://electrosam.ru/glavnaja/jelektrooborudovanie/ustrojstva/datchikitemperatury/>.

20. Датчики вологості - як вони працюють [Електронний ресурс] // Електрик Інфо. – 2009. – URL:

<http://elektrik.info/main/automation/1083-datchiki-vlazhnosti-kak-ustroeny-irabotayut.html>.

21. Датчик вологості для вентиляції [Електронний ресурс] // ELQUANTA.RU -інтернет-енциклопедія про все. – 2015. – URL: <https://elquanta.ru/novoe/ventilator-s-datchikom-vlaznosti.html>.

22. Гігрометр-психрометр віт – що це таке і для чого використовується? [Електронний ресурс] // Стимул – URL: <https://stimyl.ru/articles/gigrometr-psihrometricheskij-vit-1-i-vit-2proverennaya-vremenem-nadezhnost.html>.

23. Використання датчиків вимірювання вологості повітря [Електронний ресурс] // 7 ЛЕСТНИЦ. – 2019. – URL:

<https://7lestnic.com/bez-rubriki/vse-uzlyrezultat-klasterizaciidatchik-vlaznostivozduha-dla-upravlenia-ventilaciej.html>.

24. Барометр [Електронний ресурс] // Wikipedia. – 2021. – URL: <https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D1%80%D0%BE%D0%BC%D0%B5%D1%82%D1%80>.

25. Станційний барометр [Електронний ресурс] // Науково-популярний проект "Метеоролог і я". – 2006. – URL:

<https://meteo59.ru/book/pribory-i-nablyudeniya/barometr.php>.

26. Півпровідникові датчики температури [Електронний ресурс] // mcuspu.ru – URL:

<https://mcuspu.ru/index.php/pdevices/datchiki/108poluprovodnikovye>.

27. Барометр. Види та робота. Застосування та налаштування. Особливості [Електронний ресурс] // ТЕХ.ПРИБОРЫ.РУ. – 2021. – URL: <https://tehpribory.ru/glavnaia/pribory/barometr.html>.
28. Акустичні датчики температури [Електронний ресурс] // Познайка.Орг. – 2016. – URL: <https://poznayka.org/s75216t1.html>.
29. Створення програми автоматизованої системи визначення відносної вологості [Електронний ресурс] // Студентська бібліотека онлайн. – 2013. – URL: <https://studbooks.net/1998628/informatika/vvedenie>.
30. Метеорологічна станція [Електронний ресурс] // Вікіпедія. – 2020. – 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.
31. Автоматична дорожня метеостанція «КОНДОР» [Електронний ресурс] // Компанія ТОВ "ІТЦ-М". – 2021. – URL: <https://www.itc.by/its-solution/road-weather-station/>.
32. Метеорологічна станція: види, інструменти та прилади, що проводяться спостереження. [Електронний ресурс] // agromassidayu. – 2021. – URL: <https://ukr.agromassidayu.com/meteorologicheskaya-stanciya-vidi-instrumenti-ipribori-provodimie-nablyudeniya-a-411125#menu-6>.
33. Forestry Weather Station [Електронний ресурс] // Dyacon – URL: <https://dyacon.com/forestry-weather-station/>.
34. Modbus [Електронний ресурс] // Wikipedia. – 2021. – URL: https://ru.wikipedia.org/wiki/Modbus#%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82_MODBUS.
35. Hydrology Weather Station [Електронний ресурс] // Dyacon – URL: <https://dyacon.com/hydrology-weather-station/>.

36. Hydrometeorological Hazards: Monitoring, Forecasting, Risk Assessment, and Socioeconomic Responses [Електронний ресурс] // Huan Wu et al. – 2016. – URL: <https://www.hindawi.com/journals/amete/2016/2367939/>.
37. The SMART2000 Telemetry System [Електронний ресурс] // Felix Technology Inc. – URL:
<http://www.felixtek.com/productsdevices-and-systems/environmental-equipment/environmental-sensors-anddataloggers/smart2000-telemetry-system>.
38. Як вибрати потрібну метеостанцію [Електронний ресурс] // VENCON. – 2006. – URL: <https://vencon.ua/ua/articles/kak-vybratmeteostanciyyu>.
39. TEMPEST FORECASTING [Електронний ресурс] // WeatherFlow. – 2021. – URL: <https://weatherflow.com/tempestforecasting/>.
40. ESP8266. ПОЧАТОК РОБОТИ, ОСОБЛИВОСТІ [Електронний ресурс] // Alexgyver. – 2023. – URL: <https://alexgyver.ru/lessons/esp8266>.
41. BMP280 Digital Pressure Sensor [Електронний ресурс] // Bosch Sensortec. – 2015. – URL:
https://drive.google.com/file/d/1zow9w2wrUG9l9oskkMLlA6PZRB_FqvR/view?pli=1
42. MicroPython IDEs for ESP32 and ESP8266 [Електронний ресурс] // Randomnerdtutorials – 2022 – URL:
<https://randomnerdtutorials.com/micropython-ides-esp32-esp8266/>
43. Початок роботи з Arduino IDE та платою Arduino [Електронний ресурс] // Mikrokontroller – 2022 – URL:
<https://microkontroller.ru/arduino-projects/nachalo-raboty-s-arduino-ide-i-platoj-arduino>
44. Scratch for Arduino [Електронний ресурс] // Habr – 2013 – URL: <https://habr.com/ru/articles/196752>.
45. Документація по мові JavaScript [Електронний ресурс] // Developer – 2022 – URL: <https://developer.mozilla.org/uk/docs/Web/JavaScript>.

ДОДАТОК А

Охорона праці та безпека в надзвичайних ситуаціях

Охорона праці — це система правових, організаційно-технічних, гігієнічних, лікувально-профілактичних, соціально-економічних заходів і засобів, що забезпечують безпеку, збереження здоров'я і працездатності людини в процесі праці. Взаємини між працівниками та роботодавцем з питань безпеки, гігієни праці та виробничого середовища підприємстві регулює Закон України "Про охорону праці". Належна організація охорони праці, яка відповідає вимогам нормативно-правових актів, запобігає виробничому травматизму і професійним захворюванням.

При розробці дипломного проекту "дослідження та розробка системи автоматичного керування метеостанцією" використовується комп'ютер, локальна та глобальна мережі, при виготовленні — паяльне обладнання, флюс, припій, мультиметр. Перед нами стоїть завдання: визначити небезпечні фактори для робочого місця, а також розробити запобіжні заходи, спрямовані на уникнення будь-яких шкідливих впливів та збереження здоров'я працівників.

А.1 Загальна характеристика приміщення

Приміщення, де знаходиться обладнання, є робочою кімнатою лабораторії дослідницького інституту, яка розташована на третьому поверсі панельної будівлі. Віб्राція приміщення відсутня, склад повітря — в нормі. У кімнаті знаходиться комп'ютер, монітор, мультиметр, меблі. Покриття підлоги — паркет. Засклення листове подвійне. Колір стелі — білий, стін — світлий, підлоги — світло-коричневий. Світлові отвори орієнтовані на південь. Протилежна будівля — на відстані приблизно 100 м. В лабораторії працюють два фахівця. Мережа, що живить технічні прилади, має наступні властивості:

змінний струм частотою 50 Гц;

мережа з глухозаземленою нейтраллю;

габарити приміщення;

довжина 6 м;

ширина 4,3 м;

висота 2,6 м.

Розрахуємо площу:

$$S = l \cdot b \quad (\text{A.1})$$

де l — довжина приміщення; b — ширина приміщення.

$$S = 6 \times 4,3 = 25,8 \text{ (м}^2\text{)}$$

Об'єм приміщення:

$$V = S \cdot h \quad (\text{A.2})$$

де h — висота приміщення.

$$V = 25,8 \times 2,6 = 67,1 \text{ (м}^3\text{)}$$

Площа та об'єм меблів: $S_m \approx 5,3 \text{ (м}^2\text{)}$, $V_m \approx 8 \text{ (м}^3\text{)}$ Віднімаємо площу та об'єм меблів від розрахованих: $S = 25,8 - 5,3 = 20,5 \text{ (м}^2\text{)}$

$$V = 67,1 - 8 = 59,1 \text{ (м}^3\text{)}$$

Порівнюємо площу та об'єм приміщення в перерахунку на одну людину з нормативними показниками (табл. 1)

Таблиця А.1 – Площа і об'єм приміщення

Параметр	Норматив	Розрахункові дані
Площа	не менше 6 м ²	10,3 м ²
Об'єм	не менше 15 м ³	29,6 м ³

З цього видно, що умови праці в такому приміщенні задовольняють за площею та об'ємом на одну особу вимоги СН 245–85.

А.2 Оцінка санітарних умов праці під час паяння

Подивимося на процес паяння з точки зору майбутньої дослідницько-конструкторської розробки. Буде реалізовано процес ручного паяння за допомогою електричного паяльника безперервної дії потужністю 20–40 Вт.

В якості припою використовується олов'яно-свинцевий припій марки ПОС-63 ГОСТ 21931–76, в якості флюсу — безкислотний флюс КЕ ГОСТ 1797–64. Утворення свинцевого аерозолі становить 0,02–0,04 мг/100 пайок. Для видалення залишків флюсу застосовується етиловий спирт. До складу припою ПОС-63 входить олово в кількості 62,5–63,5% та свинець у кількості 36,5–37,5%. Флюс складається з соснової каніфолі в кількості 15–28% та етилового спирту 72–85%.

Свинець є надзвичайно небезпечною речовиною (клас 1), тому його ГДК в повітрі робочої зони становить 0,01 мг/м³. Олово, є помірно небезпечною речовиною (клас 3) і його ГДК — 10 мг/м³. Спирт етиловий є мало небезпечною речовиною (клас 4) — ГДК 1000 мг/м³.

А.3 Мікроклімат

Робота інженера не потребує фізичних навантажень, але відноситься до нервово-напруженої діяльності, тому умови мікроклімату в приміщенні відповідають нормальним значенням у відповідності до ГОСТ 12.1.005–88* та ДСН 3.3.6.042–99.

Мікроклімат підтримується у холодний період року за рахунок загальної системи опалення, а у теплий — вентиляції та кондиціонування згідно зі ДБНВ.2.5-67:2013.

А.4 Освітлення

Якість освітлення є необхідною умовою для здійснення виробничих завдань

працівниками. Воно є важливим стимулятором організму людини, а його недостатній рівень стає причиною втоми зорового аналізатора під час виконання роботи, що може посприяти травматизму. Освітлення має відповідати наступним нормативам: бути достатнім, рівномірним; не утворювати тіней на робочій поверхні; не засліплювати працівника; напрямок світлового потоку повинен відповідати зручному виконанню роботи.

Необхідно спланувати розташування робочого місця так, щоб природне світло, що надходить крізь вікна, було з лівого боку. Коефіцієнт природної освітленості має бути не нижче 1,5%. Робоче місце з ПК має розташовуватись так, щоб уникнути попадання в очі прямого світла. Джерела штучного освітлення розташовують з обох боків від екрану паралельно напрямку зору. Для уникнення блисків застосовують спеціальні фільтри для екрану, антиблискові окуляри та жалюзі на вікнах.

А.5 Оцінка природного освітлення

У приміщенні бічне однобічне природне освітлювання. У кімнаті — два вікна шириною 1,5 м та висотою 1,6 м. Місто Київ знаходиться у IV світловому поясі. Для будівель, що розташовані у цьому поясі коефіцієнт природної освітленості розраховується з формулою:

$$e_n = e^{III} \cdot m \cdot c \quad (A.3)$$

де e^{III} — ненормований коефіцієнт природної освітленості для даного виду зорових робіт (1,5% для даного виду робіт);

m — коефіцієнт світлового клімату; c — коефіцієнт сонячності клімату.

Для IV світлового поясу $t=0,9$. Значення c залежить від орієнтації світлових прорізів щодо стін горизонту (від азимуту в градусах). Зарозташування світлових прорізів в зовнішніх стінах будівель цей коефіцієнт

береться для трьох інтервалів азимута: у IV світловому поясі світлового

клімату 0,75; 0,8; 1,0 (північніше 50° північної широти) та 0,7; 0,75; 0,9 (50° північної широти південніше). Відповідно для заданих проектних умов можна прийняти $c=0,7$.

Отже, нормоване значення коефіцієнту природної освітленості: $e_n = 1,5 \cdot 0,9 \cdot 0,7 = 0,95$

Коефіцієнт запасу, що враховується при виконанні розрахунку природнього освітлення K_z .

$$K_z = 1,3 \dots 1,5.$$

Прийнято $K_z = 1,35$

Відношення довжини приміщення l до його ширини b $l/b = 6,0/4,3 = 1,4$

Відношення ширини приміщення b до висоти верхньої кромки вікна відносно робочої поверхні h b/h .

$$b/h = 4,3/2,3 = 1,9$$

Світлова характеристика вікон η_v . При $l/b = 1,4$ та $b/h = 1,9$.

$$\eta_v = 10,5$$

Коефіцієнт світлопропускання для матеріалу вікна τ_1 . Для подвійних склопакетних металопластикових вікон. $\tau_1 = 0,86$

Коефіцієнт, що враховує втрати світла у віконній рамі τ_2 . Для подвійних металопластикових рам вікна.

$$\tau_2 = 0,86$$

Коефіцієнт, що враховує втрати світла у несучих конструкціях τ_3 . При боковому освітленні.

$$\tau_3 = 1$$

Коефіцієнт, що враховує втрати світла у сонцезахисних пристроях τ_4 . Для внутрішніх вертикальних регульованих жалюзів.

$$\tau_4 = 1$$

Загальний коефіцієнт світлопропускання $\tau_{\text{заг}} = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 = 0,86 \cdot 0,86 \cdot 1 \cdot 1 = 0,74$

Коефіцієнти відбиття внутрішніх поверхонь виробничого приміщення: стелі $\rho_{\text{стелі}}$, стін $\rho_{\text{стін}}$, підлоги $\rho_{\text{підлоги}}$, %.

Визначається в залежності від стану та кольору внутрішньої поверхні: для стелі білого кольору $\rho_{\text{стелі}}=65\dots85\%$ (прийнято $\rho_{\text{стелі}}=85\%$);

для стін пофарбованих у зелений колір $\rho_{\text{стін}}=25\dots40\%$ (прийнято $\rho_{\text{стін}}=41\%$);

для підлоги $\rho_{\text{підлоги}}=10\dots40\%$ (для паркету світлого кольору $\rho_{\text{підлоги}}=38\%$).

Площі внутрішніх поверхонь виробничого приміщення: стелі $S_{\text{стелі}}$, стін $S_{\text{стін}}$, підлоги $S_{\text{підлоги}}$, м²:

$$S_{\text{стелі}}=l \cdot b=6 \cdot 4.3=25.8$$

$$S_{\text{стін}}=2 \cdot (l+b) \cdot H=2 \cdot (6+4.3) \cdot 3.2=53.6$$

$$S_{\text{підлоги}}=25.8$$

Середній коефіцієнт відбиття внутрішніх поверхонь виробничого приміщення $\rho_{\text{сер}}$

$$\rho_{\text{сер}}=(\rho_{\text{стелі}} \cdot S_{\text{стелі}}+\rho_{\text{стін}} \cdot S_{\text{стін}}+\rho_{\text{підлоги}} \cdot S_{\text{підл}}) / (S_{\text{стелі}}+S_{\text{стін}}+S_{\text{підл}}) \cdot 100\%=$$

$$=(85 \cdot 25.8)+(41 \cdot 53.6)+(38 \cdot 25.8) / (25.8+53.6+25.8) \cdot 100\%=0,51$$

Співвідношення, що визначають геометрію виробничого приміщення: l/b , b/h , a/b , де a - відстань середини робочої поверхні до зовнішньої стіни приміщення.

$$l/b=1.4; b/h=1.9; a/b=0.35;$$

Коефіцієнт, що враховує підсилення коефіцієнта природного освітлення у виробничому приміщенні завдяки світлу, що відбивається від внутрішніх поверхонь виробничого приміщення $r_1=F(\rho_{\text{сер}}, a/b, b/h, l/b)$. При $\rho_{\text{сер}}=0,51$; $l/b=1,4$; $b/h=1.9$; $a/b=0,35$.

Відношення відстані до протилежної будівлі D до висоти карнизу протилежної будівлі відносно підвіконня $H' D/H'$

$$D/H'=20/7=2.8$$

Коефіцієнт, що враховує вплив протилежної будівлі на освітленість у виробничому приміщенні $K_{\text{буд}}$. При $D/H'=2.8$

$$K_{\text{буд}} = 1.03$$

Площа вікон, що необхідна для забезпечення нормованого природного освітлення у розрахунковому виробничому приміщенні: S_v , м².

$$S_v = (e_n \cdot K_z \cdot \eta_v \cdot S_{\text{підлоги}} \cdot K_{\text{буд}}) / (\tau_z \cdot r_1 \cdot 100) = (0,95 \cdot 1,35 \cdot 10,5 \cdot 25,8 \cdot 1,03) / (1 \cdot 0,8) = 4,56$$

У виробничому приміщенні вмонтовано два вікна розмірами $s \times d = 1.5 \times 1.6$ м. Їх загальна площа складає 4,8 м².

Таким чином сумарної площі вікон вистачає для забезпечення нормативного природного освітлення у приміщенні, де відбуватиметься процес розробки та створення проекту.

А.6 Оцінка штучного освітлення

В якості джерел світла використовують люмінесцентні лампи низького тиску ЛБ60 в кількості 8 штук, розміщені в чотирьох світильниках Л201- 2×80-02, розташованих на стелі в 1 ряд.

Розрахуємо освітленість за допомогою методу коефіцієнта використання світлового потоку:

$$E = \frac{N \cdot \Phi_e \cdot \eta}{S \cdot k_z \cdot Z} \quad \text{с} \quad (\text{A.4})$$

де N — кількість світильників;

Φ — світловий потік світильника ($3 \times 2700 = 8100$ лм);

η — коефіцієнт використання світлового потоку (0,5 для забарвлення стін, стелі та підлоги);

S — площа приміщення (25.8 м²);

k — коефіцієнт запасу для даного типу ламп і приміщення (1,5);

Z — коефіцієнт нерівномірності освітлення (1,1)

Отриманий результат розрахунку дозволяє порівняти число з нормою для заданих умов:

- середній контраст об'єкту розрізнення;
- середній фон;
- розряд зорової роботи Зв;
- робота високої точності.

Для таких умов праці норма загального освітлення робочих місць має бути не менше 300 лк. Умови ДБН В.2.5.28-2018 виконані.

Місцеве освітлення

На робочому місці комбіноване освітлення повинно забезпечувати освітленість $E=750$ лк. Загальне освітлення забезпечує 380 лк, відповідно місцеве освітлення повинно дорівнювати $750-380=370$ лк.

Оскільки розраховане значення освітленості джерела місцевого сяйва перевищує мінімально необхідне 370 лк, обране джерело цілком відповідає вимогам нормативів освітленості у заданому приміщенні.

Шум та вібрація

Джерела шуму у приміщенні — вентилятори системних блоків ПЕОМ, кондиціонер та система вентиляції.

У робочому приміщенні рівень звукового тиску, рівень шуму та еквівалентні рівні шуму відповідають вимогам ДСН 3.3.6.037-99.

Рівень шуму складає 40 дБ(А), що не перевищує норму $LA = 50$ дБ(А) згідно ДСН 3.3.6.037-99. Оптимальними заходами для зниження рівня шуму є своєчасні змащення систем вентиляції системних блоків комп'ютерів.

Оскільки рівень вібрації в приміщенні незначний, засоби запобігання вібрації не розглядаються.

А.7 Електробезпека

Лабораторія за небезпекою ураження електричним струмом відноситься до приміщень з підвищеною небезпекою, оскільки можливе одночасне доторкання людини до металевих конструкцій споруд, що мають з'єднання з одного боку з ґрунтом, а з іншого – з металевим корпусом електрообладнання відповідно до ПУЕ-87.

В приміщенні наявна шина повторного заземлення, яка відповідає вимогам ПУЕ „Правила улаштування електроустановок”. (Розділ 1 Загальні правила. Гл.1.7 Заземлення і захисні заходи електробезпеки) та з'єднується із заземленою нейтраллю електроустановки.

Електробезпека електричних приладів забезпечується комплексом конструктивних, схемно-конструктивних та експлуатаційних засобів та способів захисту.

До основних експлуатаційно-технічних заходів електробезпеки відносять: захисне заземлення; занулення; вирівнювання потенціалів; малу напругу; електричне розподілення мереж (за допомогою розділових трансформаторів); захисне вимкнення; ізоляція струмопровідних частин (робоча, додаткова, підсилена, подвійна); компенсація струмів замикання на землю; огорожувальні пристрої; електрична попереджувальна сигналізація; блокування. Допоміжними технічними засобами, що використовуються при обслуговуванні та експлуатації ЕП, вважають плакати – знаки безпеки/небезпеки; засоби індивідуального захисту, запобіжне улаштування, дерев'яні решітки, гумові килими.

А.8 Пожежна безпека

Причинами виникнення пожежі в лабораторії можуть бути: несправність електропроводки, коротке замикання електричних ланцюгів, перегрівання апаратури, блискавка.

Пожежна безпека згідно з НАПБ А.01.001-2014 забезпечується системами запобігання пожежі, пожежним захистом та організаційно-технічними заходами.

З точки зору пожежної безпеки відповідно до ПУЕ клас зони приміщення П-Па (приміщення, в якому горючі речовини знаходяться у твердому або волокнистому стані без виділення пилу та волокон).

Категорія приміщення згідно з ДСТУ Б В.1.1-36:2016 за вибуховою, пожежно-вибуховою та пожежною небезпекою відноситься до категорії В пожежнооасна, оскільки в приміщенні наявні тверді горючі матеріали, здатні при взаємодії з киснем повітря або один з одним.

Ступінь вогнестійкості будівлі II згідно з ДБН В 1.1-7-2016, оскільки будівля відноситься до будинків з несучими та огорожувальними конструкціями з цегли та залізобетону.

У системі пожежного захисту передбачені аварійне виключення і перемикання апаратури, система сповіщення — звукова сигналізація.

Пожежна безпека людини забезпечується використанням вуглекислотного вогнегасника ВВК-5, ємністю 5 літрів відповідно з речовиною гасіння вогню малої електропровідності. Застосування пінних вогнегасників виключено, тому що обладнання може перебувати під напругою.

ДОДАТОК Б

Перша версія коду

```

#include <ESP8266WebServer.h>

#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_BME280.h>

#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME280 bme;

float temperature, humidity, pressure, altitude;

/* Введите SSID и пароль от вашей сети */

const char* ssid = "YourNetworkName"; // SSID
const char* password = "YourPassword"; // пароль

ESP8266WebServer server(80);

void setup()
{
  Serial.begin(115200);

  delay(100);

  bme.begin(0x76);

  Serial.println("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);

    Serial.print(".");

  }

  Serial.println("");

  Serial.println("WiFi connected..!");

  Serial.print("Got IP: "); Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);

  server.onNotFound(handle_NotFound);

  server.begin();

  Serial.println("HTTP server started");
}

```

```

void loop()
{
    server.handleClient();
}

void handle_OnConnect()
{
    temperature = bme.readTemperature();
    humidity = bme.readHumidity();
    pressure = bme.readPressure() / 100.0F;
    altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
    server.send(200, "text/html", SendHTML(temperature,humidity,pressure,altitude));
}

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

String SendHTML(float temperature,float humidity,float pressure,float altitude)
{
    String ptr = "<!DOCTYPE html> <html>\n";

    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";

    ptr += "<title>ESP8266 Weather Station</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>ESP8266 Weather Station</h1>\n";

    ptr += "<p>Temperature: ";

    ptr += temperature;

    ptr += "&deg;C</p>";

    ptr += "<p>Humidity: ";

    ptr += humidity;

```

```
ptr += "%</p>";  
ptr += "<p>Pressure: ";  
ptr += pressure;  
ptr += "hPa</p>";  
ptr += "<p>Altitude: ";  
ptr += altitude;  
ptr += "m</p>";  
ptr += "</div>\n";  
ptr += "</body>\n";  
ptr += "</html>\n";  
return ptr;}
```

ДОДАТОК В

Друга версія коду

```

#include <ESP8266WebServer.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme;
float temperature, humidity, pressure, altitude;
const char* ssid = "Wi-Fi25_10"; // SSID
const char* password = "42091982"; // пароль
ESP8266WebServer server(80);

void setup()
{
  Serial.begin(115200);
  delay(100);
  bme.begin(0x76);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());
  server.on("/",handle_OnConnect);
  server.onNotFound(handle_NotFound);
  server.begin();
  Serial.println("HTTP server started");

}

void loop()

```

```

{
    server.handleClient();
}

void handle_OnConnect()
{
    temperature = bme.readTemperature();
    humidity = bme.readHumidity();
    pressure = bme.readPressure() / 100.0F;
    altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
    server.send(200, "text/html", SendHTML(temperature,humidity,pressure,altitude));
}

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

String SendHTML(float temperature,float humidity,float pressure,float altitude)
{
    String ptr = "<!DOCTYPE html>";
    ptr += "<html>";
    ptr += "<head>";
    ptr += "<title>ESP8266 Weather Station</title>";
    ptr += "<meta name='viewport' content='width=device-width, initial-scale=1.0'>";
    ptr += "<link href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600'";
    ptr += "rel='stylesheet'>";
    ptr += "<style>";
    ptr += "html { font-family: 'Open Sans', sans-serif; display: block; margin: 0px auto; text-align: center;color: #444444;}";
    ptr += "body{margin: 0px;} ";
    ptr += "h1 {margin: 50px auto 30px;} ";
    ptr += ".side-by-side{display: table-cell;vertical-align: middle;position: relative;}";
    ptr += ".text{font-weight: 600;font-size: 19px;width: 200px;}";
    ptr += ".reading{font-weight: 300;font-size: 50px;padding-right: 25px;}";
    ptr += ".temperature .reading{color: #F29C1F;}";
    ptr += ".humidity .reading{color: #3B97D3;}";
    ptr += ".pressure .reading{color: #26B99A;}";
    ptr += ".altitude .reading{color: #955BA5;}";

```

```

ptr += ".superscript{font-size: 17px;font-weight: 600;position: absolute;top: 10px;}";
ptr += ".data{padding: 10px;}";
ptr += ".container{display: table;margin: 0 auto;}";
ptr += ".icon{width:65px}";
ptr += "</style>";
ptr += "<script>\n";
ptr += "setInterval(loadDoc,1000);\n";
ptr += "function loadDoc() {\n";
ptr += "var xhttp = new XMLHttpRequest();\n";
ptr += "xhttp.onreadystatechange = function() {\n";
ptr += "if (this.readyState == 4 && this.status == 200) {\n";
ptr += "document.body.innerHTML =this.responseText}\n";
ptr += "};\n";
ptr += "xhttp.open(\"GET\", \"/\", true);\n";
ptr += "xhttp.send();\n";
ptr += "}\n";
ptr += "</script>\n";

ptr += "</head>";
ptr += "<body>";
ptr += "<h1>ESP8266 Weather Station</h1>";
ptr += "<div class='container'>";
ptr += "<div class='data temperature'>";
ptr += "<div class='side-by-side icon'>";

ptr += "<svg enable-background='new 0 0 19.438 54.003'height=54.003px id=Layer_1 version=1.1
viewBox='0 0 19.438 54.003'width=19.438px x=0px xml:space=preserve xmlns=http://www.w3.org/2000/svg
xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path d='M11.976,8.82v-
2h4.084V6.063C16.06,2.715,13.345,0,9.996,0H9.313C5.965,0,3.252,2.715,3.252,6.063v30.982";

ptr += "C1.261,38.825,0,41.403,0,44.286c0,5.367,4.351,9.718,9.719,9.718c5.368,0,9.719-4.351,9.719-
9.718";

ptr += "c0-2.943-1.312-5.574-3.378-7.355V18.436h-3.914v-2h3.914v-2.808h-4.084v-2h4.084V8.82H11.976z
M15.302,44.833";

ptr += "c0,3.083-2.5,5.583-5.583,5.583s-5.583-2.5-5.583-5.583c0-2.279,1.368-4.236,3.326-
5.104V24.257C7.462,23.01,8.472,22,9.719,22";

ptr += "s2.257,1.01,2.257,2.257V39.73C13.934,40.597,15.302,42.554,15.302,44.833z'fill=#F29C21
/></g></svg>";

ptr += "</div>";
ptr += "<div class='side-by-side text'>Temperature</div>";
ptr += "<div class='side-by-side reading'>";

```

```

ptr +=(int)temperature;

ptr +="<span class='superscript'>&deg;C</span></div>";

ptr +="</div>";

ptr +="<div class='data humidity'>";

ptr +="<div class='side-by-side icon'>";

ptr +="<svg enable-background='new 0 0 29.235 40.64'height=40.64px id=Layer_1 version=1.1
viewBox='0 0 29.235 40.64'width=29.235px x=0px xml:space=preserve xmlns=http://www.w3.org/2000/svg
xmlns:xlink=http://www.w3.org/1999/xlink y=0px><path
d='M14.618,0C14.618,0,0,17.95,0,26.022C0,34.096,6.544,40.64,14.618,40.64s14.617-6.544,14.617-
14.617";

ptr +="C29.235,17.95,14.618,0,14.618,0z M13.667,37.135c-5.604,0-10.162-4.56-10.162-10.162c0-
0.787,0.638-1.426,1.426-1.426";

ptr
+="c0.787,0,1.425,0.639,1.425,1.426c0,4.031,3.28,7.312,7.311,7.312c0.787,0,1.425,0.638,1.425,1.425";

ptr +="C15.093,36.497,14.455,37.135,13.667,37.135z'fill=#3C97D3 /></svg>";

ptr +="</div>";

ptr +="<div class='side-by-side text'>Humidity</div>";

ptr +="<div class='side-by-side reading'>";

ptr +=(int)humidity;

ptr +="<span class='superscript'>%</span></div>";

ptr +="</div>";

ptr +="<div class='data pressure'>";

ptr +="<div class='side-by-side icon'>";

ptr +="<svg enable-background='new 0 0 40.542 40.541'height=40.541px id=Layer_1 version=1.1
viewBox='0 0 40.542 40.541'width=40.542px x=0px xml:space=preserve xmlns=http://www.w3.org/2000/svg
xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path d='M34.313,20.271c0-0.552,0.447-1,1-
1h5.178c-0.236-4.841-2.163-9.228-5.214-12.593l-3.425,3.424";

ptr +="c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293c-0.391-0.391-0.391-1.023,0-
1.414l3.425-3.424";

ptr +="c-3.375-3.059-7.776-4.987-12.634-5.215c0.015,0.067,0.041,0.13,0.041,0.202v4.687c0,0.552-
0.447,1-1,1s-1-0.448-1-1v0.25";

ptr +="c0-0.071,0.026-0.134,0.041-
0.202C14.39,0.279,9.936,2.256,6.544,5.385l3.576,3.577c0.391,0.391,0.391,1.024,0,1.414";

ptr +="c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293L5.142,6.812c-2.98,3.348-
4.858,7.682-5.092,12.459h4.804";

ptr +="c0.552,0,1,0.448,1,1s-0.448,1-
1,1H0.05c0.525,10.728,9.362,19.271,20.22,19.271c10.857,0,19.696-8.543,20.22-19.271h-5.178";

ptr +="C34.76,21.271,34.313,20.823,34.313,20.271z M23.084,22.037c-0.559,1.561-2.274,2.372-
3.833,1.814";

ptr +="c-1.561-0.557-2.373-2.272-1.815-3.833c0.372-1.041,1.263-1.737,2.277-
1.928L25.2,7.202L22.497,19.05";

ptr +="C23.196,19.843,23.464,20.973,23.084,22.037z'fill=#26B999 /></g></svg>";

```

```

ptr += "</div>";

ptr += "<div class='side-by-side text'>Pressure</div>";

ptr += "<div class='side-by-side reading'>";

ptr += (int)pressure;

ptr += "<span class='superscript'>hPa</span></div>";

ptr += "</div>";

ptr += "<div class='data altitude'>";

ptr += "<div class='side-by-side icon'>";

ptr += "<svg enable-background='new 0 0 58.422 40.639' height=40.639px id=Layer_1 version=1.1
viewBox='0 0 58.422 40.639' width=58.422px x=0px xml:space=preserve xmlns=http://www.w3.org/2000/svg
xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path d='M58.203,37.754l0.007-0.004l42.09,9.935l-
0.001,0.001c-0.356-0.543-0.969-0.902-1.667-0.902";

ptr += "c-0.655,0-1.231,0.32-1.595,0.808l-0.011-0.007l-0.039,0.067c-0.021,0.03-0.035,0.063-
0.054,0.094l22.78,37.692l0.008,0.004";

ptr += "c-0.149,0.28-0.242,0.594-
0.242,0.934c0,1.102,0.894,1.995,1.994,1.995v0.015h31.888c1.101,0,1.994-0.893,1.994-1.994";

ptr += "C58.422,38.323,58.339,38.024,58.203,37.754z' fill=#955BA5 /><path d='M19.704,38.674l-0.013-
0.004l13.544-23.522l25.13,1.156l-0.002,0.001C24.671,0.459,23.885,0,22.985,0";

ptr += "c-0.84,0-1.582,0.41-2.051,1.038l-0.016-0.01l20.87,1.114c-0.025,0.039-0.046,0.082-
0.068,0.124l0.299,36.851l0.013,0.004";

ptr += "C0.117,37.215,0,37.62,0,38.059c0,1.412,1.147,2.565,2.565,2.565v0.015h16.989c-0.091-0.256-
0.149-0.526-0.149-0.813";

ptr += "C19.405,39.407,19.518,39.019,19.704,38.674z' fill=#955BA5 /></g></svg>";

ptr += "</div>";

ptr += "<div class='side-by-side text'>Altitude</div>";

ptr += "<div class='side-by-side reading'>";

ptr += (int)altitude;

ptr += "<span class='superscript'>m</span></div>";

ptr += "</div>";

ptr += "</div>";

ptr += "</body>";

ptr += "</html>";

return ptr;
}

```