

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КВАЛІФІКАЦІЙНОЇ
РОБОТИ БАКАЛАВРА

на тему:

**«Пристрій вимірювання температури і вологості повітря
середовища на базі плат управління NodeMCU V3»**

Завідувач кафедри ЕіКТ

А.С. Опанасюк

Керівник роботи

А.С. Опанасюк

Студент групи ЕС-71

І.С. Ротенберг

РЕФЕРАТ

Пояснювальна записка: 41 аркуш; 32 рисунків; 10 джерел.

Графічні матеріали: 1 принципова електрична схема, 1 функціональна схема, 1 схема алгоритму, 1 структурна схема.

Об'єкт розробки – пристрій вимірювання температури та вологості оточуючого повітря.

Мета роботи розробка приладу, який при підключенні до інтернет мережі робить вимірювання та передає данні про такі характеристики повітря, як температура і вологість.

Перший розділ є оглядом літературних джерел, він містить постановку завдання, тут розглядаються аналоги до кожного з елементів компонентної бази, їх особливості та технічні характеристики .

Другий розділ містить інформацію про розробку алгоритму роботи та структурної схеми програмно-апаратного комплексу.

Третій розділ включає в себе розробку та опис функціональної схеми мікропроцесорного пристрою контролю у складі програмно-апаратного комплексу.

Четвертий розділ містить розробку та опис принципової електричної схеми мікропроцесорного пристрою контролю

П'ятий розділ містить опис процесу збірки, налаштування, прошивки приладу. Тут також представлений повний програмний код, що використовується для управління логічними процесами приладу.

Ключові слова: ESP8266, NodeMCU, DHT22, ArduinoIDE, HTML, температура, вологість, SSID, IoT.

ЗМІСТ

ЗМІСТ.....	1
Вступ.....	3
1 Огляд літератури та постановка задачі	4
1.1 Аналіз програмно-апаратних продуктів-аналогів	4
1.2 Постановка задачі проектування.....	9
2 Розроблення алгоритму роботи та структурної схеми програмно-апаратного комплексу.....	11
2.1 Синтез алгоритму розробки та функціонування програмно-апаратного комплексу.....	11
2.2 Розроблення структурної схеми пристрою вимірювання температури і вологості повітря.....	15
3 Розроблення та опис схеми функціональної мікропроцесорного блоку ESP8266	17
4 Розроблення та опис схеми принципово електричної мікропроцесорного блоку ESP8266.....	24
4.1 Інтерфейс вводу-виводу загального призначення (GPIO).....	24
4.2 Інтерфейс Secure Digital Input / Output (SDIO).....	24
4.3 Послідовний інтерфейс периферійних пристроїв (SPI / HSPI) ..	25
4.4 Інтерфейс I2C.....	26
4.5 Інтерфейс I2S.....	26

					ЕЛІТ 6.171.00.10.462 ПЗ				
Зм.	Лист	№ докум.	Підпис	Дата	Пристрій вимірювання температури і вологості повітря середовища на базі плат управління NodeMCU V3.	Літ.	Аркуш	Аркушів	
Розроб.		Ропендберг І.С.					1	41	
Перевір.		Опанасюк А.С.							
Т. Контр.									
Н. Контр.									
Затверд.		Опанасюк А.С.			СумДУ, зр. ЕС-71				

4.6	Універсальний асинхронний прийомопередавач (UART).....	27
4.7	Широтно-імпульсна модуляція (PWM).	28
4.8	ІК пульт дистанційного керування.....	29
4.9	Світлодіоди і кнопки.....	30
4.10	Принципово електрична схема.....	30
5	Процес збірки та прошивки приладу.....	32
5.1	Підключення плати управління до датчику температури та вологості. 32	
5.2	Програмування та налаштування пристрою.	34
	Висновки	41
	Список літератури.....	42

ВСТУП

В наш час IoT (Internet of Things) прилади стрімко набувають популярності серед населення світу. Глобальне розповсюдження доступу до інтернет мережі сприяло зростанню зацікавленості середньостатистичного користувача у приладах і системах, які можуть полегшити його рутинні процеси, використовуючи дротові та бездротові протоколи інтернет з'єднання.

Метою роботи є розробка приладу, який при підключенні до інтернет мережі міг би робити вимірювання та передавати данні про такі характеристики повітря, як температура і вологість. Завдяки розробленому пристрою, користувач зможе здійснювати моніторинг параметрів повітря, наприклад, у будь якого місці своєї оселі, знаходячись при цьому на роботі, в дорозі, або в будь-якій точці планети, де є точка доступу до мережі інтернет. Також приладом можуть скористатися аграрії або фермери, які зможуть здійснювати моніторинг температури і вологості у місцях утримання великої рогатої худоби, теплицях, сараях тощо. Відображення одержаних даних буде відбуватися на портативному або стаціонарному комп'ютерному пристрої користувача в форматі HTML в режимі реального часу (з поправкою на латентність датчика). Розроблений пристрій може використовуватися, як в стаціонарному так і портативному режимі завдяки підключенню зовнішнього енергоносія.

При розробці пристрою в якості елементної бази були відібрані наступні модулі: плата управління NodeMCU V3 на базі мікропроцесора ESP8266 та датчик температури і вологості повітря DHT22. В процесі створення проекту було розроблене програмне забезпечення для коректного функціонування та виконання всіх інструкцій і алгоритмів збору, обробки та відправки даних.

Базуючись на вище викладеній інформації, а також дослідженні актуальності описаного приладу, був спроектований та розроблений пристрій вимірювання температури та вологості оточуючого повітря [\[1\]](#).

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		3

1 ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз програмно-апаратних продуктів аналогів

При дослідженні теми був зроблений висновок, що на споживчому ринку відсутні аналоги розглянутого пристрою. Тому було прийнято рішення зробити порівняння елементної бази даного проекту. Пунктами порівняння будуть ключові елементи пристрою, а точніше плата управління NodeMCU V3, а також датчик DHT11. У аналоги обраної плати управління можна записати наступне, плати: ESP8266 NodeMCU V2, NodeMCU V3, ESP32 NodeMCU, ESP8266 WeMos D1 Mini, Arduino NANO 3, Arduino UNO R3, Arduino UNO WIFI R2, Arduino Mega. Для більш комфортного порівняння цих елементів можна використовувати таблицю зображену на рисунку 1.1.

	ESP8266 NodeMCU V2	NodeMCU V3	ESP32 NodeMCU	ESP8266 WeMos D1 Mini	Arduino NANO 3	Arduino UNO R3	Arduino UNO WIFI R2	Arduino Mega
Microcontroller	ESP8266	ESP8266	ESP32	ESP8266	ATmega328p	ATmega328p	ATmega4809	ATmega2560
Operating Voltage	3.3V	3.3V	3.3V	3.3V	5V	5V	5V	5V
Power supply	7V – 12V	7V – 12V	7V – 12V	4V – 6V	7V – 12V	7V – 12V	7V – 12V	7V – 12V
Current consumption	15 μ A – 400 mA	15 μ A – 400 mA	20 mA – 240 mA		19 mA – 180 mA	45 mA – 80 mA	50 mA – 150 mA	50 mA – 200 mA
Current consumption Deep Sleep	0.5 μ A	0.5 μ A	5 μ A		23 μ A (with special settings)	35 mA	35 mA	500 μ A
Analog Input Pins	1	1	15	1	8	6	6	16
DC Current per I/O Pin	12 mA	12 mA	20 mA		40 mA	40 mA	40 mA	20 mA
Flash Memory	4 MB	4 MB	4 MB	4 MB	32 KB	32 KB	48 KB	256 KB
SRAM	n.A	64 KB	520 KB		2 KB	2 KB	6 KB	8 KB
EEPROM	512 bytes	512 bytes	–		1024 bytes	1024 bytes	256 bytes	4096 bytes
Clock Speed	52 MHz	80 MHz	80 MHz / 160 MHz	80 MHz / 160 MHz	16 MHz	16 MHz	16 MHz	16 MHz
Price	\$6	\$6.50	\$11	\$6	\$22	\$22	\$44.90	\$38.50
WIFI	yes	yes	yes	yes	no	no	yes	no

Рисунок 1.1 – Таблиця порівняння технічних характеристик плат управління NodeMCU та Arduino.

З цієї таблиці видно, що обрана нами плата управління є відмінним варіантом для розглянутого проекту, оскільки має оптимальний баланс між ціною, продуктивністю і набором функцій. Таким чином, ми розуміємо, що за ціновий діапазон в районі 6\$ ми отримуємо один з найбільш продуктивних мікропроцесорів серед аналогів (80 MHz в порівнянні з 16 MHz у більшості конкурентів), а також інтегрований набір бездротових функцій. Не кажучи вже про значну кількість постійної і оперативної пам'яті (4 MB і 64 KB відповідно, в порівнянні з 32 KB і 2 KB відповідно у молодших версій Arduino), що дає нам в перспективі високу гнучкість під час додавання функціональності в даний пристрій, якщо у кінцевого споживача з'явиться така необхідність. Також не варто забувати про підсумкову вартість пристрою, оскільки при виборі плати під брендом Arduino навіть найбільш базові їх версії коштуватимуть в рази дорожче, ніж обрана нами NodeMCU v3. Саме з огляду на сукупності всіх цих фактів було прийнято рішення використовувати в якості плати управління NodeMCU v3, бо вона ідеально збігається з критеріями проекту, а саме, зібрати пристрій з необхідним кінцевому споживачеві функціоналом за мінімальну вартість елементної бази [2].

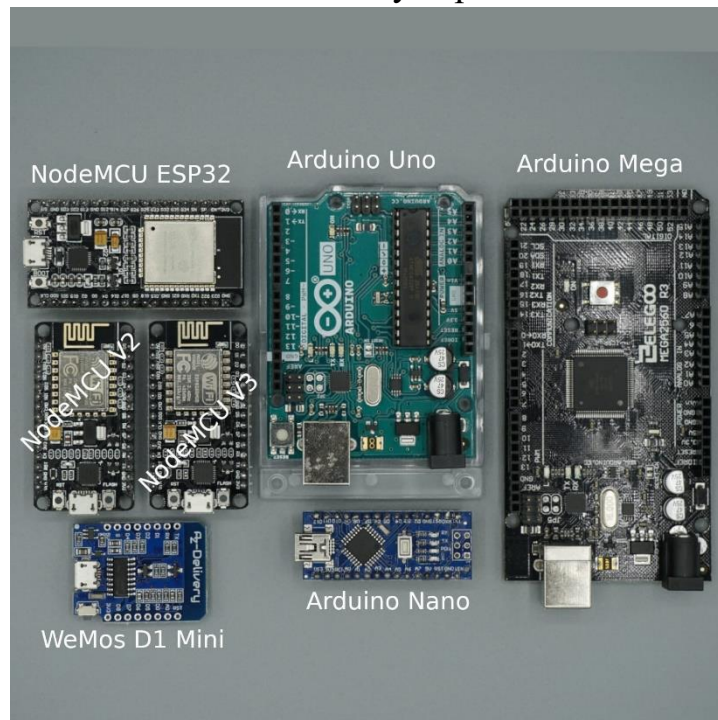


Рисунок 1.2 – Фотографія зовнішнього вигляду та топографії плат управління які беруть участь у порівнянні

						ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм..	Лист	№ докум.	Підпис	Дата			5

Нижче перейдемо до порівняння обраного нами датчика температури і вологості з його найближчими аналогами. Як аналоги для датчика DHT22 було обрано такі моделі: DHT11, LM35, DS18B20, BME280, BMP180. Для спрощення процесу порівняння технічних характеристик, нижче наведена таблиця (Рис. 1.3)







						
Sensor	DHT11	DHT22 (AM2302)	LM35	DS18B20	BME280	BMP180
Measures	Temperature Humidity	Temperature Humidity	Temperature	Temperature	Temperature Humidity Pressure	Temperature Pressure
Communication protocol	One-wire	One-wire	Analog	One-wire	I2C SPI	I2C
Supply voltage	3 to 5.5V DC	3 to 6V DC	4 to 30 V DC	3 to 5.5V DC	1.7 to 3.6V (for the chip) 3.3 to 5V for the board	1.8 to 3.6V (for the chip) 3.3 to 5V for the board
Temperature range	0 to 50°C	-40 to 80°C	-55 to 150°C	-55 to 125°C	-40 to 85°C	0 to 65°C
Accuracy	+/- 2°C (at 0 to 50°C)	+/- 0.5°C (at -40 to 80°C)	+/-0.5°C (at 25°C)	+/-0.5°C (at -10 to 85°C)	+/-0.5°C (at 25°C)	+/-0.5°C (at 25°C)

Рисунок 1.3 – Порівняння технічних характеристик обраних моделей датчиків

Спираючись на свідчення наведені на рис. 1.3 ми можемо зробити висновок, що обраний нами датчик температури і вологості, є найбільш оптимальним варіантом для використання в нашій системі оскільки він в порівнянні з його найближчими аналогами має найбільш прийнятне співвідношення технічних характеристик, простотою конструкції і ціни. Наприклад, якщо розглядати обраний нами датчик DHT22 і його аналог від того ж виробника DHT11, то ми помітимо, що у аналога температурний діапазон вимірювань, коливається від 0 до 50 °С що є недостатнім показником якщо розглядати погодні умови, які превалюють, в середині зимового сезону в Україні. Також не викликає довіри верхній поріг температури в 50 °С, оскільки значення температури повітря в середині літнього сезону можуть досягати 40 і більше градусів Цельсія. З огляду на значення, отримані під час практичного випробування розглянутих датчиків, які наведені у вигляді графіка (рис 1.4), ми можемо зробити висновок що даний датчик може видавати досить неточний результат при вимірюваннях в умовах підвищеної температури навколишнього середовища.

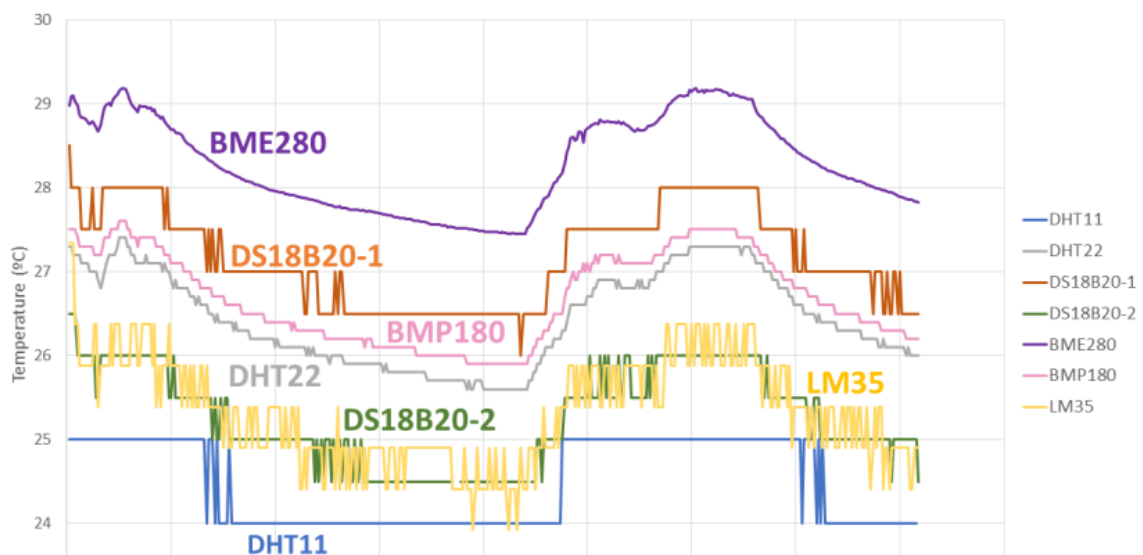


Рисунок 1.4 – Графік даних отриманих при випробуванні датчиків аналогів

Ще одним цікавим конкурентом для нашого порівняння можна назвати датчик LM35. Його температурний діапазон роботи і загальна простота конструкції підключення може підкупити, але у цього датчика є кілька критичних особливостей через які ми не можемо використовувати його в нашому проекті. Перша особливість, на яку варто звернути увагу найбільше, це той факт, що він повністю аналоговий. Як результат можемо зробити висновок що при використанні цього датчика ми будемо мати проблеми з налаштуванням і оптимізацією його роботи при підключенні до нашої плати управління. Наприклад, для датчика DHT22 є готові розроблені бібліотеки, в той час як для датчика LM35 з огляду на його аналогову конструкцію доведеться повністю з нуля прописувати алгоритм базової інтеграції функцій, що є безглуздою тратою часу і сил. Другою критичною особливістю є його похибка, оскільки вказано що його похибка в 0,5 градусів є актуальною тільки при середньому значенні температури навколишнього середовища в 25 градусів Цельсія, Даний факт вказує на те, що при віддаленні від середнього показника температури ми будемо мати все більшу похибку у вимірюваннях даного датчика. І нарешті під кінець обговорення хочеться додати, що температурний діапазон від -55 °C і аж до 150 °C не має сенсу, оскільки передбачені нами сценарії використання не передбачають знаходження пристрою в умовах занадто низьких або занадто високих температур [3].

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

1.2 Постановка задачі проектування

Метою даної кваліфікаційної роботи бакалавра є розробка програмно-апаратного комплексу вимірювання стану температури та вологості повітря в умовах приміщення, критої зони або відкритого простору. Пріоритетом є зручність та швидкість доступу до значень мікроклімату у місці знаходження приладу.

Комплекс повинен відповідати таким вимогам та мати:

1. Високу точність вимірювання кліматичних показників, похибка вимірювань не повинна перевищувати поріг в 0,5 градусів Цельсія.
2. Конструкція пристрою повинна передбачати не великі розміри для комфорту транспортування та встановлення на місці функціонування.
3. Можливість підключення до стаціонарних або портативних джерел енергоживлення з вихідною напругою 5 В, а також параметром струму від 600 мА.
4. Частота опитування датчику повинна бути не більше 2 с. для актуалізації інформації в режимі реального часу.
5. Відправка пакетів даних до статичного веб серверу здійснюється у момент отримання інформації з датчика, тому на транслявання даних повинно бути виділено вікно у 5 мс.
6. Зручність та простоту користування, наявність візуально зрозумілого графічного інтерфейсу з використанням піктограмних елементів.
7. Можливість відображення даних у будь-який момент часу на портативному або стаціонарному комп'ютерному пристрої який має можливість підключення до мережі і відображення HTML інформації.

В ході розробки, для вирішення поставлених задач було використано об'єктно-орієнтоване програмування мовою C++, середовище розробки Microsoft Visual Studio (рис. 1.5) [\[4\]](#).

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		9

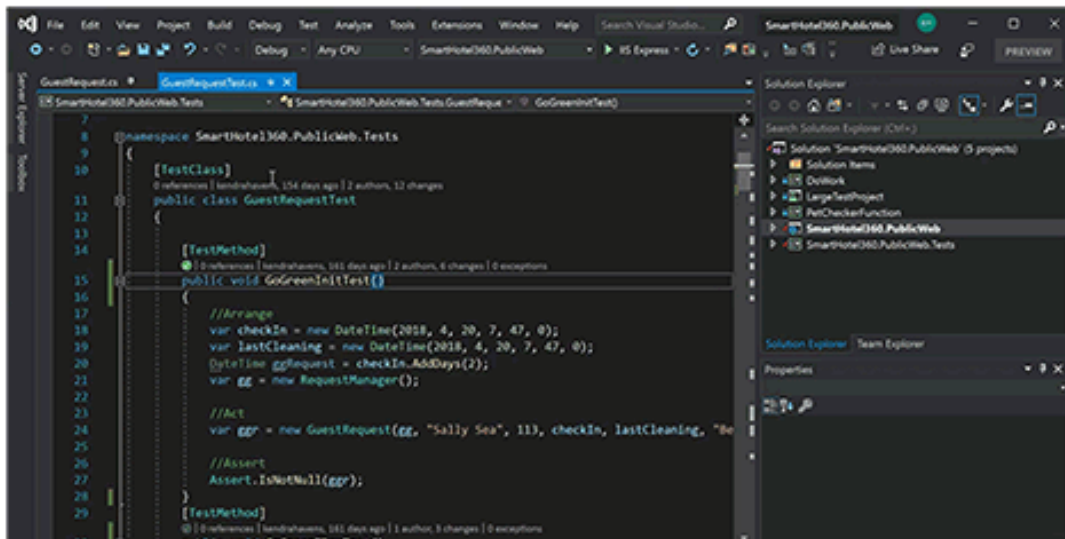


Рисунок 1.5 – Середовище програмування Microsoft Visual Studio 2019

Для роботи з написаним кодом та його подальшого запису на внутрішнє постійне сховище плати управління NodeMCU v3 було використано програмне середовище ArduinoIDE (рис. 1.6) [5].

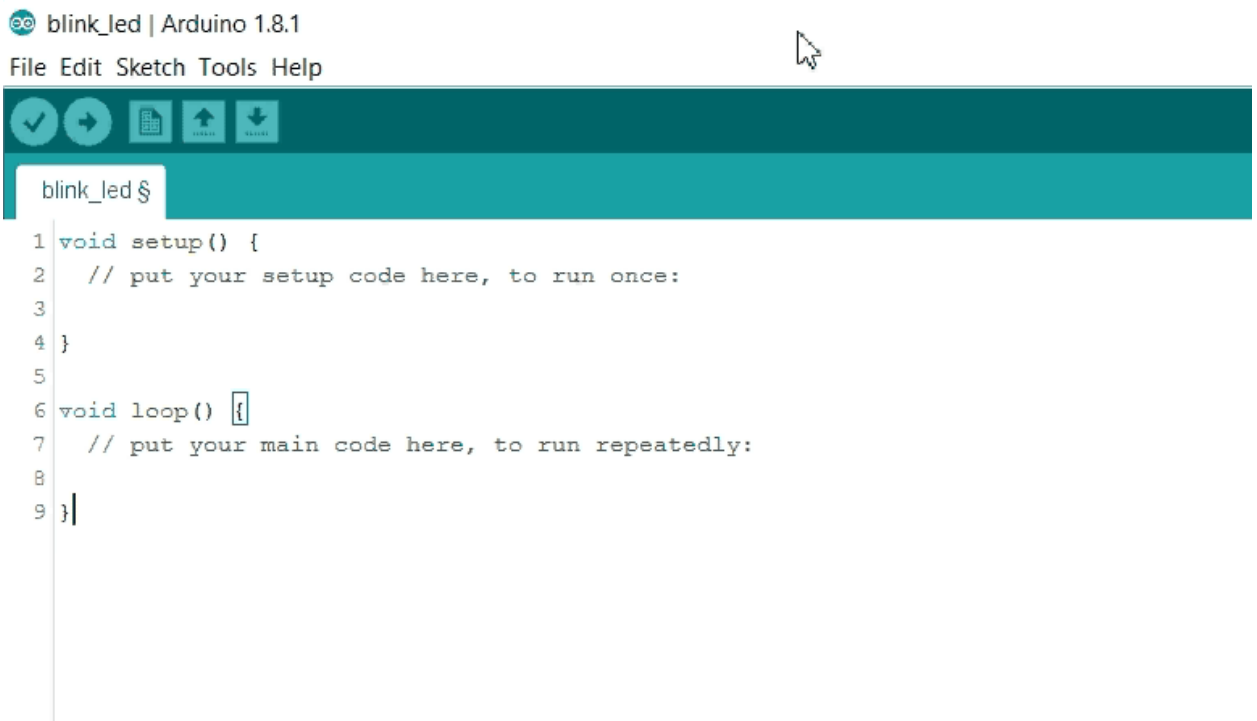


Рисунок 1.6 – Програмне середовище ArduinoIDE

2 РОЗРОБЛЕННЯ АЛГОРИТМУ РОБОТИ ТА СТРУКТУРНОЇ СХЕМИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

2.1 Синтез алгоритму розробки та функціонування програмно-апаратного комплексу

Для коректного налаштування та програмування плати управління та кліматичного датчика було створено блок-схему алгоритму функціонування портативної метеостанції. Ця блок-схема включає в себе наступні ключові елементи:

1. Тип датчику, SSID та пароль мережі.
2. Налаштування HTTP серверу.
3. Підключення до мережі.
4. Перевірка достовірності даних.
5. Зчитування даних з датчиків.
6. Відображення та стилізація HTML сторінки.

На першому етапі нам потрібно визначити номер відводу ESP8266 NodeMCU, до якого підключений вивід даних датчика, і створити об'єкт DHT, щоб можна було отримати доступ до спеціальних функцій, пов'язаних з бібліотекою DHT. У нашому випадку ми будемо використовувати інформаційний пін під кодовою назвою D8.

Одразу після етапу ідентифікації інформаційного піну плата управління, за допомогою підключеної бібліотеки даних, визначає тип та модель датчику, який у даний момент підключений до інформаційного піну. Також цю інформацію можливо одразу прописати у алгоритмічних даних, цей варіант підходить нам набагато краще оскільки ми не плануємо використовувати інші датчики окрім обраного нами в цьому проекті.

На наступному етапі іде перевірка SSID та паролю мережі, ці данні були попередньо записані в постійну пам'ять плати управління як частина первинного коду з інструкціями що до послідовного виконання операцій.

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

Наступним кроком іде процес налаштування HTTP серверу. Реалізувати цю процедуру ми будемо через підключення бібліотеки ESP8266WebServer.h що має кілька методів, які допоможуть нам налаштувати сервер і обробляти вхідні HTTP запити, не турбуючись про деталі низькорівневої реалізації. Задля реалізації серверного базису ми оголошуємо об'єкт бібліотеки ESP8266WebServer, щоб отримати доступ до її функцій. Конструктор цього об'єкта в якості параметра приймає номер порту (який сервер буде прослуховувати). Ми використаємо порт 80 оскільки це порт за замовчуванням для HTTP. Тепер ми можемо отримати доступ до сервера, не вказуючи порт в URL.

Одразу після цього етапу змінні типу float, а саме Temperature і Humidity оголошуються для зберігання значень відповідно температури і вологості.

Четвертим етапом є створення і наповнення функції setup (). В цій функції прописується процес перевірки достовірності даних. Перш за все для налагодження потрібно відкрити послідовне з'єднання і встановити порти GPIO на вхід. Також нам потрібно формувати об'єкт DHT за допомогою функції begin (). Тепер нам потрібно підключитися до мережі Wi-Fi за допомогою функції WiFi.begin (). Ця функція приймає в якості параметрів SSID (ім'я мережі) і пароль. Поки ESP8266 буде намагається підключитися до мережі, нам потрібно буде перевірити стан підключення за допомогою функції WiFi.status (). Як тільки ESP8266 буде підключений до мережі, скетч надрукує IP адрес, присвоєний для ESP8266, відображаючи значення WiFi.localIP () в моніторі послідовного порту. Щоб обробляти вхідні HTTP запити, нам потрібно вказати, який код виконувати при введенні певного URL. Для цього ми використовуємо метод on (). Цей метод приймає два параметри. Перший - це URL шлях, а другий - ім'я функції, яку ми хочемо виконати при вході на цей URL. Якщо клієнт буде запитувати будь-який URL, що відрізняється від зазначених в server.on () він повинен видати відповідь з HTTP статусом 404 (Not Found, «сторінка не знайдена») і повідомленням для користувача. Для цього потрібно розмістити ці дії в окрему функцію і використати метод server.onNotFound (), щоб вказати серверу, що він повинен виконати, коли отримає запит на URI, який не був заданий за допомогою server.on (). Для того щоб запустити сервер в цілях перевірки або експлуатації потрібно викликати метод begin () для об'єкта server.

						ЕлІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата			12

П'ятим етапом є задання процесу зчитування даних з датчиків. Щоб обробити реальні вхідні HTTP запити, потрібно буде викликати метод `handleClient ()` об'єкта `server`. Далі потрібно буде створити функцію, яку ми прикріпили до кореневого (`/`) URL за допомогою `server.on ()`. Щоб відповісти на HTTP запит, використовуємо метод `send ()`. Хоча цей метод можна викликати з іншим набором аргументів, його найпростіша форма складається з HTTP коду відповіді. У нашому випадку ми оберемо і відправимо код 200 (один з кодів стану HTTP), який відповідає відповіді ОК. Потім потрібно вказати тип контенту як `"text / html"`, і, в кінці, ми викликаємо для користувача функцію `SendHTML ()`, яка створює динамічну HTML сторінку, що містить значення температури і вологості.

Останнім етапом є відображення та стилізація HTML сторінки. Функція `SendHTML ()` відповідає за створення веб-сторінки щоразу, коли веб-сервер ESP8266 отримує запит від веб-клієнта. Вона просто об'єднує HTML код у великий рядок і повертає її в функцію `server.send ()`. Ця функція в якості параметрів для динамічного генерування HTML контенту приймає значення температури і вологості. Перший текст, який ми повинні відправити, - це оголошення `<!DOCTYPE>`, яке вказує, що ми відправляємо HTML код. Потім `<meta>` елемент `viewport` робить веб-сторінку адаптивною в будь-якому веб-браузері. Далі тег `title` встановлює заголовок сторінки. Далі йде робота з мовою програмування CSS для створення стилізованого зовнішнього вигляду веб-сторінки. Ми вибираємо шрифт `Helvetica`, визначаємо контент, який буде відображатися у вигляді `inline-block` і вирівняний по центру. Наступним кроком потрібно встановити, шрифт і поля навколо елементів `body`, `H1`, `H3` і `p`. Далі встановлюємо заголовок на веб-сторінці. Щоб динамічно відобразити значення температури і вологості, ми поміщаємо ці значення в тег параграфа. Ці значення перетворюються в ціле число за допомогою приведення типів. Для відображення символу градуса ми використовуємо HTML сутність `& deg;`.

Блок-схему алгоритму роботи програмно-апаратного комплексу зображено на рисунку 2.1.

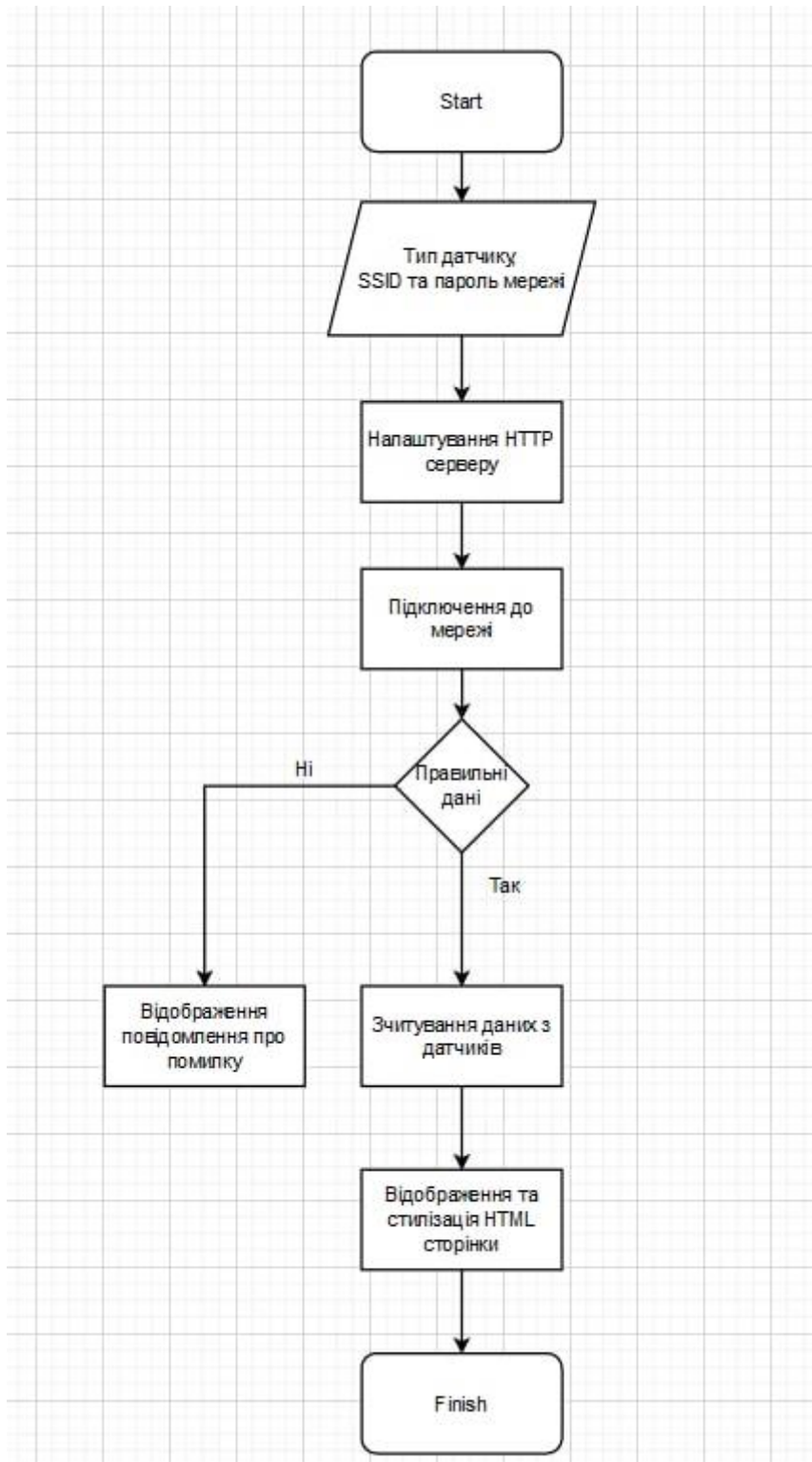


Рисунок 2.1 – Алгоритм функціонування програмно-апаратного комплексу

2.2 Розроблення структурної схеми пристрою вимірювання температури і вологості повітря.

У завершеному вигляді структурна схема розробленого пристрою складається з наступних ключових елементів:

1. Блок живлення (стаціонарний або портативний).
2. Micro-USB 2.0 type-B порт.
3. Мостовий контролер CP2102.
4. Модуль ESP-12E, що містить чіп ESP8266.
5. Модуль датчика вимірювання температури і вологості повітря DHT22.

Блок живлення в готовому пристрої повинен мати вихідні характеристики які відповідають напрузі 3.3 В і як мінімум 600 мА струму.

Порт Micro-USB використовується для програмування плати управління а також для її підключення до блоку живлення на стадії тестування та експлуатації.

Чіп CP2102 являється мостовим контролером, він перетворює USB-сигнал на послідовний і дозволяє комп'ютеру програмувати та взаємодіяти з мікросхемою ESP8266.

Модуль ESP-12E, містить мікросхему ESP8266, що має 32-розрядний RISC мікропроцесор Tensilica Xtensa® LX106, який працює на частоті від 80 до 160 МГц. Він також має 128 КБ оперативної пам'яті та 4 МБ флеш-пам'яті (для зберігання програм та даних), це достатньо для того, щоб впоратися з великими рядками, що складають веб-сторінки та дані JSON / XML. ESP8266 має інтегрований 802.11b / g / n HT40 Wi-Fi трансивер, тому він може не тільки підключатися до мережі Wi-Fi та взаємодіяти з Інтернетом, але також може створити власну мережу, що дозволяє іншим пристроям підключатися безпосередньо до нього. Це робить ESP8266 NodeMCU дуже гнучким у використанні в різного роду проектах [\[6\]](#).

DHT22 - недорогий датчик вимірювання вологості та температури. Він складається з термістору для вимірювання температури та ємнісного датчика вологості. Особливість, яка виділяє цей датчик з поміж інших, полягає в тому, що він може транслювати дані через кожні дві секунди.

Структурна схема вимірювального приладу приведена на рисунку 2.2.



Рисунок 2.2 – Структурна схема вимірювального приладу

3 РОЗРОБЛЕННЯ ТА ОПИС СХЕМИ ФУНКЦІОНАЛЬНОЇ МІКРОПРОЦЕСОРНОГО БЛОКУ ESP8266

В ході роботи над проектом була розроблена та описана функціональна схема мікропроцесорної системи ESP8266 також відомої під робочою назвою ESP8266EX. Далі буде розписана інформація що до функціоналу, призначення та характеристик функціональних блоків і комплексів.

ESP8266EX містить 32-розрядний мікропроцесор Tensilica lx106. Тактова частота процесора - 80 МГц, можливо програмне переключення в режим 160 МГц. Система може легко працювати як операційна система реального часу (RTOS). Wi-Fi стек займає тільки 20% процесорного часу. Решту процесорного часу може бути використана для додатків користувача. Мікропроцесор може взаємодіяти з іншими частинами чіпу через такі інтерфейси:

- Програмований RAM / ROM інтерфейс (iBus), з'єднаний з контролером пам'яті і має доступ до зовнішньої флеш-пам'яті.
- Інтерфейс пам'яті даних (dBus), з'єднаний з контролером пам'яті.
- АНВ інтерфейс, що надає доступ до периферійних регістрів.

ESP8266EX Wi-Fi SoC містить контролер пам'яті і блоки пам'яті, включаючи SRAM (ОЗУ) і ROM (ПЗУ). Мікроконтролер може звертатися до блоків пам'яті через iBus, dBus і АНВ інтерфейси. При роботі в режимі WiFi STA, обсяг оперативної пам'яті, доступної користувачу, становить 50 КБ. Цей обсяг включає в себе стек, сегменти data і heap. У ESP8266EX немає програмованої ROM, тому програма зберігається у зовнішній SPI флеш-пам'яті.

Зовнішня SPI флеш-пам'ять, підключена до ESP8266EX, використовується для зберігання програм користувача. Теоретично підтримується до 16 Мбайт пам'яті. Опції вибору обсягу SPI флеш-пам'яті:

- Без використання OTA: мінімальний розмір флеш-пам'яті - 512 Кб.
- З використанням OTA: мінімальний розмір флеш-пам'яті - 1 МБ.

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17

До шині АНВ крім арбітра підключені МАС, SDIO (хост) і процесор. Залежно від адреси, запити даних можуть перейти з АНВ в один з двох підлеглих блоків:

- Шину APB.
- Контролер флеш-пам'яті (як правило, для автономних додатків).

Як правило, запити до контролера флеш-пам'яті - високошвидкісні запити, а до блоку APB - доступ до регістрів. Блок APB виступає в якості декодера, який звертається до програмованим регістрів в основних блоках ESP8266EX. Залежно від адреси, запит APB може надійти в радіо, I2S, SPI, SDIO (хост), GPIO, UART, таймер реального часу (RTC), МАС або модем.

Сигнал тактової частоти ESP8266EX використовується прийомними і передавальними змішувачами. Тактовий сигнал створюється вбудованим генератором з використанням зовнішнього кварцового резонатора. Генератор може працювати з резонаторами в діапазоні від 26 МГц до 52 МГц (рис. 3.1). Внутрішнє калібрування кварцового генератора гарантує, що можна використовувати широкий спектр кварцових резонаторів. Проте, якість кварцу залишається важливим фактором, що впливає на рівень фазового шуму і чутливість Wi-Fi.

Параметр	Обозначение	Мин.	Макс.	Ед.изм.
Частота	FXO	26	52	МГц
Емкость нагрузки	CL	-	32	пФ
Динамическая емкость	CM	2	5	пФ
Сопротивление потерь	RS	0	65	Ом
Допустимое отклонение частоты	Δ FXO	-15	15	%
Частота от температуры (-25°C ~ 75°C)	Δ FXO,Temp	-15	15	%

Рисунок 3.1 – Технічні характеристики тактового генератора.

Допускається використання зовнішнього тактового генератора. Частота може бути в діапазоні від 26 МГц до 52 МГц. При наступних характеристиках (рис. 3.2) передбачається досягнення хороших показників радіопередачі.

Параметр	Обозначение	Мин.	Макс.	Ед.изм.
Амплитуда тактового сигнала	VXO	0.2	1	В
Точность внешнего крацевого резонатора	ΔFXO,EXT	-15	15	%
Фазовый шум при смещении 1 кГц, тактовой частоте 40 МГц	-	-	-120	дБ/Гц
Фазовый шум при смещении 10 кГц, тактовой частоте 40 МГц	-	-	-130	дБ/Гц
Фазовый шум при смещении 100 кГц, тактовой частоте 40 МГц	-	-	-138	дБ/Гц

Рисунок 3.2 – Вимоги до зовнішнього кварцового генератора

Приймач ESP8266EX складається з наступних блоків:

- Приймач діапазону 2,4 ГГц
- Передавач діапазону 2,4 ГГц
- Генератор високої частоти
- Таймер реального часу
- Генератор зміщення і регулятори напруги
- Управління енергоспоживанням

В наявності підтримка чотирнадцяти каналів відповідно до стандартів IEEE802.11 b / g / n (Рис. 3.3).

Но. канала	Частота (МГц)	Но. канала	Частота (МГц)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

Рисунок 3.3 – Частотні канали для стандартів IEEE802.11 b / g / n

Приймач на 2,4 ГГц перетворює радіочастотні сигнали в квадратурні сигнали проміжної частоти, які згодом оцифровуються двома високошвидкісними АЦП з високою роздільною здатністю. Для адаптації до постійно змінюваних станів каналу в ESP8266EX інтегровані радіочастотні фільтри, автоматичне регулювання посилення (AGC), ланцюги компенсації постійного зміщення та фільтри проміжної частоти.

Передавач перетворить квадратурні сигнали проміжної частоти в діапазон 2,4 ГГц, і подає високочастотний сигнал на вхід КМОП-підсилювача потужності. Функція цифрового калібрування покращує лінійність підсилювача потужності, що дозволяє досягти кращих в своєму класі значень вихідної потужності: +19,5 дБм в режимі 802.11b і +16 дБм в режимі 802.11n. В передавач інтегровані додаткові механізми калібровки, що дозволяють компенсувати недоліки радіо, такі як:

- Розсіювання несучої частоти.
- Фазову синхронізацію I/Q.
- Нелінійність трактів проміжної частоти.

Високочастотний генератор генерує квадратурні сигнали 2,4 ГГц для приймача і передавача. Всі компоненти генератора інтегровані на чіпі, в тому числі всі індуктори, варикапи, фільтри, регулятори і дільники. Генератор має вбудовані механізми калібрування і схеми самотестування. Підстроювання фаз квадратурного сигналу і зменшення фазового шуму здійснюється на чіпі запатентованими алгоритмами калібрування. Це забезпечує найкращу роботу приймача і передавача.

					ЕЛІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		20

У ESP8266EX реалізований повний 802.11 b/g/n/e/i WLAN MAC протокол з підтримкою специфікації Wi-Fi Direct. Можлива робота не тільки в режимі BSS, але і робота в групі P2P відповідно до останнього існуючого протоколу Wi-Fi P2P. Всі низькорівневі функції Wi-Fi протоколу обробляються ESP8266EX автоматично:

- RTS / CTS
- підтвердження
- фрагментація і дефрагментація
- агрегація
- інкапсуляція фреймів (802.11h / RFC 1 042)
- автоматичні моніторинг / сканування beacon-пакетів
- P2P Wi-Fi Direct.

Пасивне або активне сканування, а також процедура виявлення P2P виконується автономно після першого ініціювання відповідної команди. Управління живленням здійснюється з мінімальною взаємодією з хостом, щоб мінімізувати час роботи в активному режимі.

ESP8266 спеціально розроблений для мобільних пристроїв, переносної електроніки і сфери Інтернету речей із застосуванням передових технологій управління живленням. Передбачені 3 режиму енергоспоживання: активний режим (active mode), сплячий режим (sleep mode) і режим глибокого сну (deep sleep mode). ESP8266EX споживає близько 20 мкА в режимі глибокого сну (з працюючим таймером реального часу), і менше 1,0 мА (DTIM = 3) або менше 0.6 мА (DTIM = 10), щоб підтримувати з'єднання з точкою доступу [\[7\]](#).

- Off: напруга на виході CHIP_PU низька. Таймер реального часу відключений. Всі регістри скинуті.
- Deep sleep: Тільки таймер реального часу включений - інша частина чіпу вимкнена. Резервна пам'ять таймера реального часу може містити базову інформацію про Wi-Fi з'єднання.
- Sleep: Тільки таймер реального часу працює. Кварцовий генератор відключений. Будь-які події, які ініціюють пробудження (MAC, хост, таймер реального часу, зовнішні переривання) переведуть чіп в режим Wake up.
- Wake up: У цьому стані система переходить зі стану сну в режим On. Кварцовий генератор і ФАПЧ включені.
- On: Тактовий генератор високої частоти включений і подає сигнал на інші блоки відповідно до регістру управління тактовим сигналом. Управління подачею тактового сигналу також реалізовано всередині окремих блоків. Наприклад, подача тактового сигналу на процесор може бути припинена за допомогою інструкції WAITI.

Блок-схема управління енергоживленням зображена на рисунку 3.4.

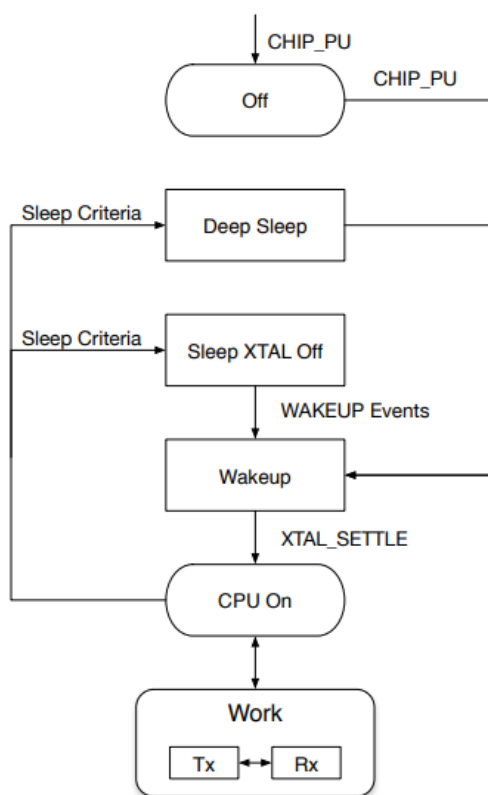


Рисунок 3.4 – Схема управління енергоживленням.

На основі всієї вище зазначеної інформації була розроблена та скомпонована функціональна схема мікропроцесорної системи (Рис 3.5).

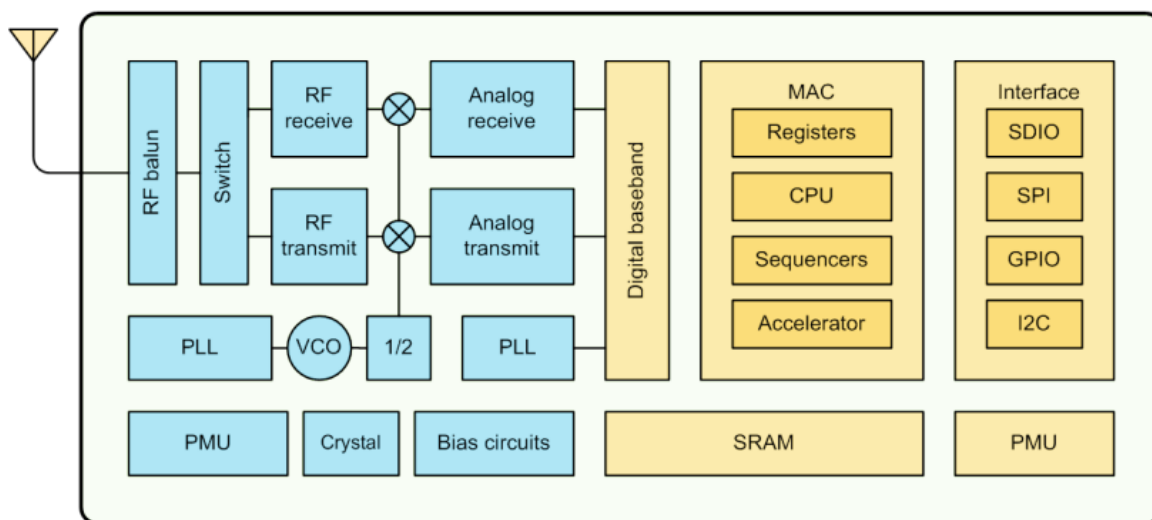


Рисунок 3.5 – Функціональна схема мікропроцесорної системи

4 РОЗРОБЛЕННЯ ТА ОПИС СХЕМИ ПРИНЦИПОВО ЕЛЕКТРИЧНОЇ МІКРОПРОЦЕСОРНОГО БЛОКУ ESP8266

4.1 Інтерфейс вводу-виводу загального призначення (GPIO)

ESP8266EX має 17 виводів GPIO, які можуть бути призначені для різних функцій шляхом програмування відповідних регістрів. Кожен GPIO є двонаправленим і неінвертованим, має внутрішні підтяжки на напругу живлення або на землю, і може бути переведений в стан з високим опором. Коли GPIO налаштований як вхід, вхідні значення можуть бути прочитані з регістрів. Вхід також може бути використаний як джерело переривань за рівнем або перепаду напруги. Ці виводи можуть бути об'єднані з іншими функціями, такими як I2C, I2S, UART, PWM, IR, і т.д. Для роботи в режимі енергозбереження виводи GPIO можуть підтримувати свій стан. Наприклад, коли чіп вимкнений, всі виходи можуть бути відключені. Додаткова функція утримання при необхідності може бути вбудована в IO. Коли IO не керується внутрішньою або зовнішньою схемою, функція утримання може бути використана для збереження останнього використаного стану. Функція утримання дає деякий позитивний зворотний зв'язок на контактну площадку. В зв'язку з цим, зовнішня схема керуюча контактом, повинна подавати струм більший, ніж струм зворотного зв'язку. Необхідна сила струму невелика - в межах 5 мкА.

4.2 Інтерфейс Secure Digital Input / Output (SDIO)

ESP8266EX має один SDIO (рис. 4.1), що працює в режимі slave, опис якого дано нижче. При цьому підтримуються 4-бітний SDIO v1.1 з частотою 25 МГц і 4-бітний SDIO v2.0 з частотою 50 МГц.

Наименование вывода	Номер вывода	IO	Функционал
SDIO_CLK	21	IO6	SDIO_CLK
SDIO_DATA0	22	IO7	SDIO_DATA0
SDIO_DATA1	23	IO8	SDIO_DATA1
SDIO_DATA_2	18	IO9	SDIO_DATA_2
SDIO_DATA_3	19	IO10	SDIO_DATA_3
SDIO_CMD	20	IO11	SDIO_CMD

Рисунок 4.1 – Опис виводів SDIO.

4.3 Послідовний інтерфейс периферійних пристроїв (SPI / HSPI)

ESP8266EX має один загальний Slave / Master SPI, один Slave SDIO / SPI, і один загальний Slave / Master HSPI. Функції всіх їх контактів можуть бути реалізовані за допомогою апаратних засобів. Опис контактів наведено нижче (рис. 4.2 - 4.4).

Наименование вывода	Номер вывода	IO	Функционал
SDIO_CLK	21	IO6	SPICLK
SDIO_DATA0	22	IO7	SPIQ/MISO
SDIO_DATA1	23	IO8	SPID/MOSI
SDIO_DATA_2	18	IO9	SPIHD
SDIO_DATA_3	19	IO10	SPIWP
SDIO_CMD	20	IO11	SPICS0
U0TXD	26	IO1	SPICS1
GPIO0	15	IO0	SPICS2

Рисунок 4.2 – Опис виводів SPI

Наименование вывода	Номер вывода	IO	Функционал
SDIO_CLK	21	IO6	SPI_SLAVE_CLK
SDIO_DATA0	22	IO7	SPI_SLAVE_MISO
SDIO_DATA1	23	IO8	SPI_SLAVE_INT
SDIO_DATA_2	18	IO9	NC
SDIO_DATA_3	19	IO10	SPI_SLAVE_CS
SDIO_CMD	20	IO11	SPI_SLAVE_MOSI

Рисунок 4.3 – Опис виводів SDIO/SPI (Slave)

Наименование вывода	Номер вывода	IO	Функционал
MTMS	9	IO14	HSPICLK
MTDI	10	IO12	HSPIQ/MISO
MTCK	12	IO13	HSPID/MOSI
MTDO	13	IO15	HPSICS

Рисунок 4.4 – Опис виводів HSPI (Slave)

4.4 Інтерфейс I2C

ESP8266EX має один I2C інтерфейс, що використовується для підключення мікроконтролерів і інших зовнішніх пристроїв, наприклад, датчиків. Опис висновків I2C представлено нижче.

Наименование вывода	Номер вывода	IO	Функционал
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Рисунок 4.5 – Опис виводів I2C

Підтримуються і I2C Master, і I2C Slave. Функціонал I2C інтерфейсу реалізований в програмному забезпеченні, максимальна тактова частота - 100 кГц. Слід зазначити, що тактова частота I2C повинна бути вище, ніж мінімальна тактова частота підлеглого пристрою.

4.5 Інтерфейс I2S

ESP8266EX має один I2S інтерфейс введення даних і один I2S інтерфейс виведення даних. I2S інтерфейси в основному використовуються в додатках для збору даних, обробки і передачі аудіо даних, а також введення і виведення послідовно переданих даних (наприклад, підтримка світлодіодів серії WS2812). Опис виводів I2S представлено нижче.

Функціонал I2S інтерфейсу (рис. 4.6) може бути здійснений за допомогою програмного забезпечення з використанням мультиплексних GPIO, пов'язаний список DMA підтримується.

I2S ввода даних			
Наименование вывода	Номер вывода	IO	Функционал
MTDI	10	IO12	I2SI_DATA
MTCK	12	IO13	I2SI_BCK
MTMS	9	IO14	I2SI_WS
MTDO	13	IO15	I2SO_BCK
U0RXD	25	IO3	I2SO_DATA
GPIO2	14	IO2	I2SO_WS

Рисунок 4.6 – Опис виводів I2S

4.6 Універсальний асинхронний прийомо-передавач (UART)

ESP8266EX має два UART інтерфейси: UART0 і UART, їх описи представлені нижче (рис. 4.7).

Тип вывода	Наименование вывода	Номер вывода	IO	Функционал
UART0	U0RXD	25	IO3	U0RXD
	U0TXD	26	IO1	U0TXD
	MTDO	13	IO15	U0RTS
	MTCK	12	IO13	U0CTS
UART1	GPIO2	14	IO2	U1TXD
	SD_D1	23	IO8	U1RXD

Рисунок 4.7 – Опис виводів UART

Передача даних в / з UART інтерфейсів може бути реалізована за допомогою апаратних засобів. Швидкість передачі даних через інтерфейси UART досягає 115200 x 40 (4,5 Мбіт / с).

UART0 може бути використаний для зв'язку. Він підтримує управління потоком. Оскільки UART1 має тільки сигнал передачі даних (Tx), то він зазвичай використовується для виведення налагоджувальної інформації.

4.7 Широтно-імпульсна модуляція (PWM).

ESP8266EX має PWM інтерфейс на чотири виводи, але вони можуть бути розширені. Опис виводів PWM інтерфейсу представлено нижче (рис. 4.8).

Наименование вывода	Номер вывода	IO	Функционал
MTDI	10	IO12	PWM0
MTDO	13	IO15	PWM1
MTMS	9	IO14	PWM2
GPIO4	16	IO4	PWM3

Рисунок 4.8 – Опис виводів PWM інтерфейсу.

Функціонал PWM інтерфейсів може бути реалізований за допомогою програмного забезпечення. Наприклад, в демонстрації розумного освітлення функція PWM здійснюється шляхом переривання таймера, мінімальне розрізнення досягає цілих 44 нс. Діапазон частот PWM регулюється в межах від 1000 мкс до 10000 мкс, тобто, між 100 Гц і 1 кГц. Коли частота PWM дорівнює 1 кГц, коефіцієнт заповнення буде дорівнювати $1/22727$, і дозвіл більше 14 біт буде досягнуто при частоті оновлення 1 кГц.

4.8 ІК пульт дистанційного керування.

Нижче представлено опис виводів одного інтерфейсу ІК ДУ (рис. 4.9).

Наименование вывода	Номер вывода	ІО	Функционал
MTMS	9	IO14	IR Tx
GPIO5	24	IO5	IR Rx

Рисунок 4.9 – Опис виводів інтерфейсу ІЧ ДК пульта дистанційного керування

Інтерфейс інфрачервоного пульта дистанційного керування (з модуляцією і демодуляцією) реалізується програмно. Інтерфейс використовує NEC кодування. Частота модульованого несучого сигналу становить 38 кГц. Дальність передачі становить близько 1 м, що визначається двома факторами: перший - максимальне значення номінального струму, другий - значення внутрішнього струмообмежувального опору в ІК-приймачі. Чим більше значення опору, тим менше струм, а відповідно і потужність, і навпаки. Кут передачі знаходиться в проміжку між 15 ° і 30 °, що визначається напрямом випромінювання інфрачервоного приймача.

4.9 Світлодіоди і кнопки

ESP8266EX має 17 GPIO, які можуть використовуватися для підтримки різних функцій світлодіодів і кнопок. Опис деяких GPIO, яким призначені певні функції, наведені нижче (рис. 4.10).

Наименование вывода	Номер вывода	IO	Функционал
MTCK	12	IO13	Button (Reset)
GPIO0	15	IO0	Wi-Fi Light
MTDI	10	IO12	Link Light

Рисунок 4.10 – Опис виводів інтерфейсу ІЧ ДК пульта дистанційного.

В цілому визначені три інтерфейси, один - для кнопки, два - для світлодіодів. Як правило, MTCK використовується для управління кнопкою скидання, GPIO0 використовується в якості сигналу для індикації робочого стану Wi-Fi, MTDI використовується в якості світлового сигналу для індикації стану зв'язку між пристроєм і сервером [8].

4.10 Принципова електрична схема

В результаті всієї вище наведеної інформації була спроектована принципово електрична схема мікропроцесорного блоку ESP8266EX (рис. 4.11)

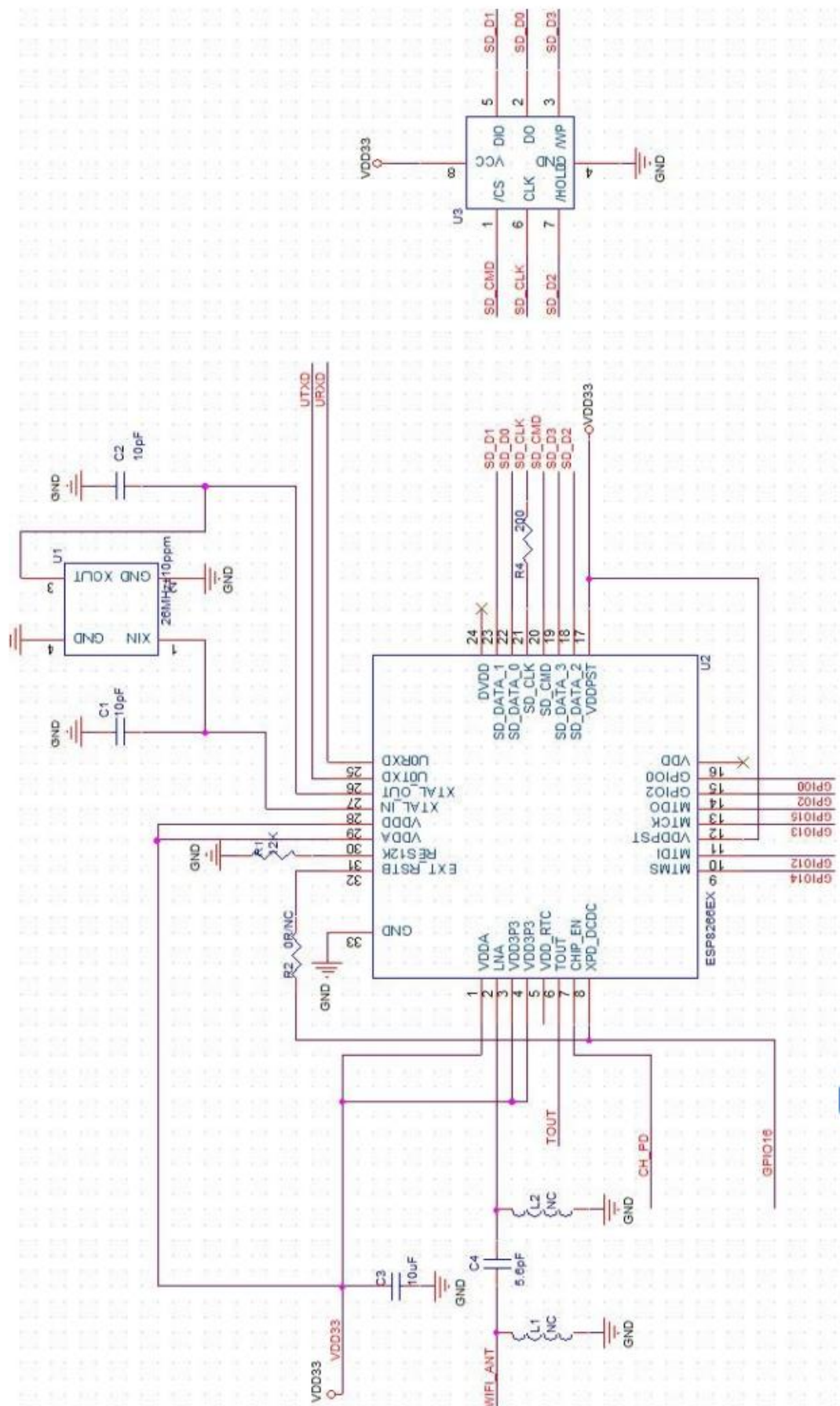


Рисунок 4.11 – Принципово електрична схема мікропроцесорного блоку ESP8266EX

Зм.	Лист	№ докум.	Підпис	Дата
-----	------	----------	--------	------

5 ПРОЦЕС ЗБІРКИ, ТА ПРОШИВКИ ПРИЛАДУ

5.1 Підключення плати управління до датчику температури та вологості

Перш за все потрібно встановити NodeMCU v3 на макетну плату, в стороні від DHT 22 (рис. 5.1). Для цього потрібно підключити вивід VCC датчика до виводу 3,3V на NodeMCU і з'єднати виводи землі датчика і плати. Далі потрібно підключити вивід даних датчика до контакту D8 плати ESP8266 NodeMCU. У нашому випадку датчик встановлений на окрему плату, тому нам не потрібно додавати будь-які зовнішні підлаштувальні резистори. Обраний нами датчик поставляється вже з вбудованим резистором на 10 кОм [\[9\]](#).

Фотографія зібраного приладу наведена на рис. 5.2.

					ЕліТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

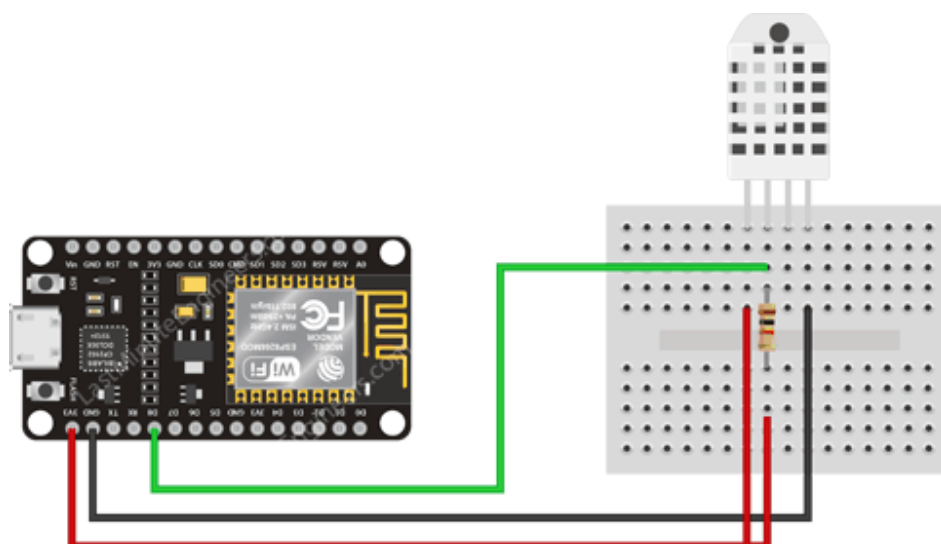


Рисунок 5.1 – Підключення плати управління до датчику

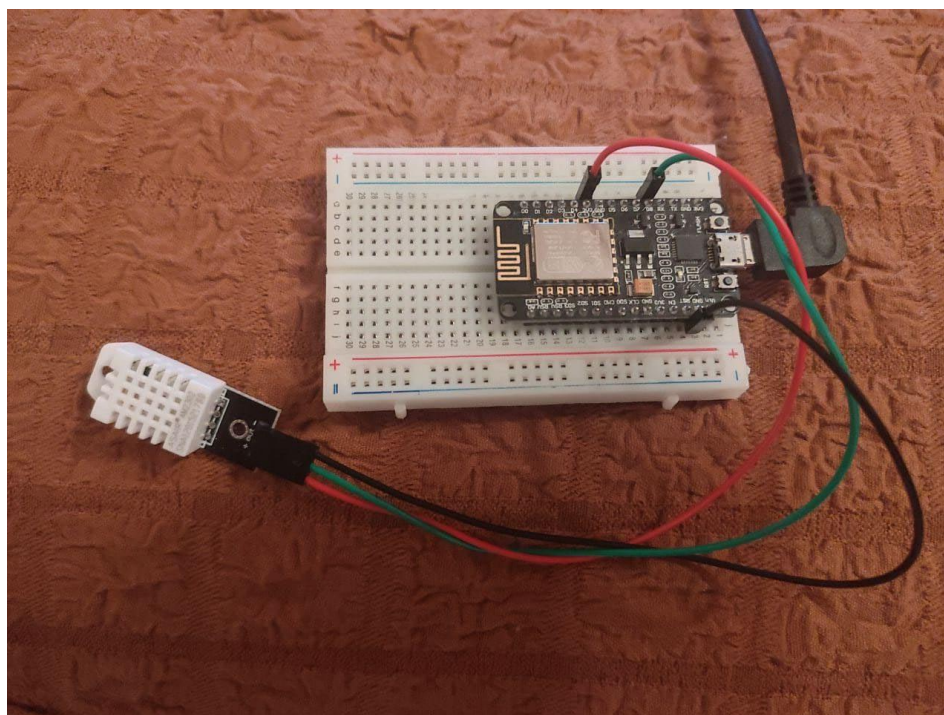


Рисунок 5.2 – Фотографія зібраного приладу

5.2 Програмування та налаштування пристрою

Датчик DHT22 для передачі даних використовує власний однодротовий протокол. Цей протокол вимагає точної синхронізації. Для полегшення роботи та вирішення цієї задачі ми будемо використовувати бібліотеку DHT від Adafruit. Ця бібліотека настільки потужна, що працює як на архітектурі Arduino (рис. 5.3), так і на архітектурі ESP.

Щоб встановити бібліотеку, нам потрібно перейти в меню Скетч (Sketch) → Підключити бібліотеку (Include Library) → Керувати бібліотеками ... (Manage Libraries ...). Зачекати, поки менеджер бібліотек завантажить індекс бібліотек і оновить список встановлених бібліотек.

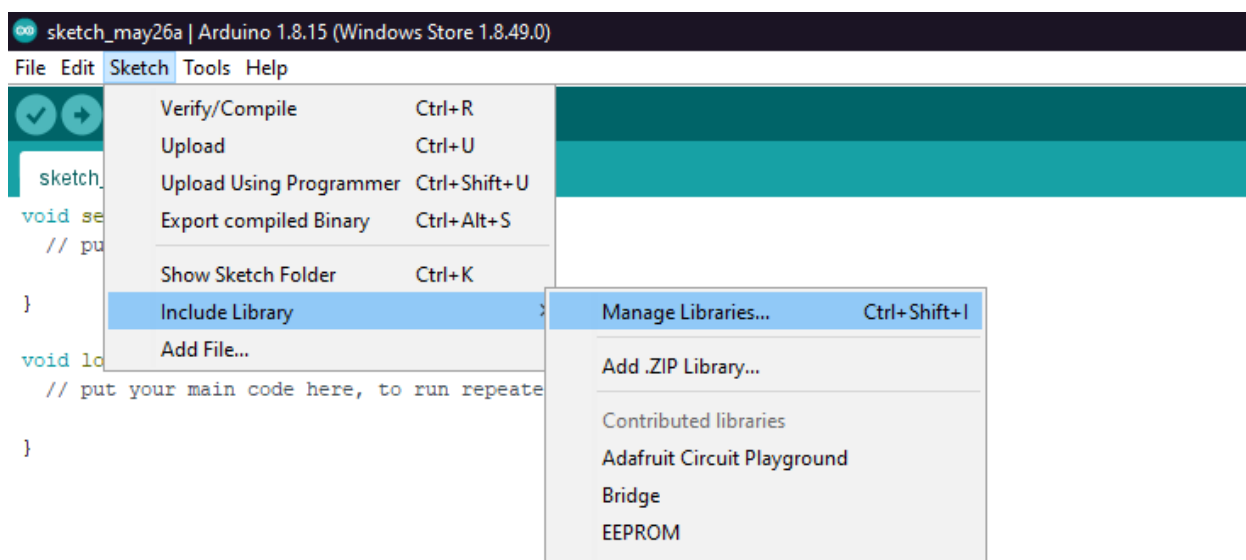


Рисунок 5.3 – Вибір управління бібліотеками в Arduino IDE

Тепер потрібно відфільтрувати результати пошуку, ввівши «DHT sensor». Знаходимо DHT sensor library by Adafruit. Натискаємо на цей пункт, а потім обираємо Встановити (Install) (рис. 5.4).

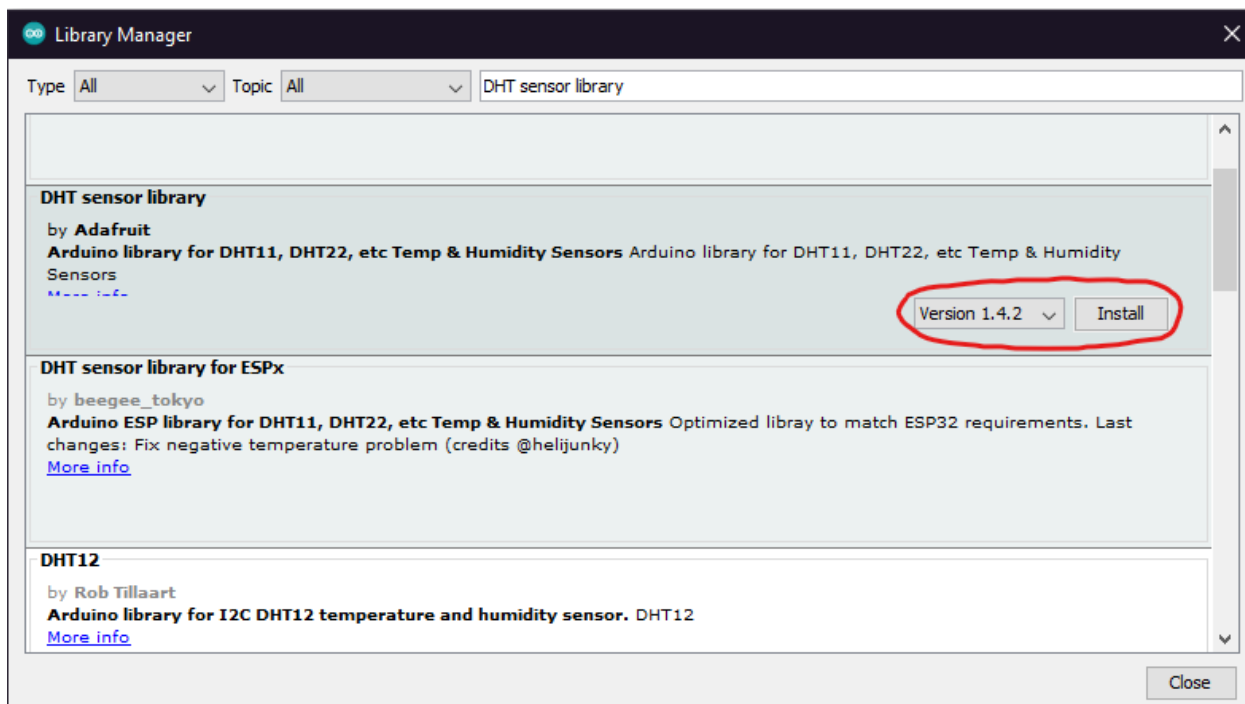


Рисунок 5.4 – Встановлення бібліотеки Adafruit DHT.

Бібліотека датчиків DHT використовує Adafruit Sensor support backend. Тому при інсталяції бібліотеки для DHT програма одразу сповістить про те що обрана бібліотека використовує ще одну додаткову бібліотеку, і одразу запропонує інсталювати і її теж. Нас це влаштовує, тому ми погоджуємося інсталювати додаткову бібліотеку через зображене діалогове вікно, натиснувши пункт Інсталювати все (Install all) (рис. 5.5).

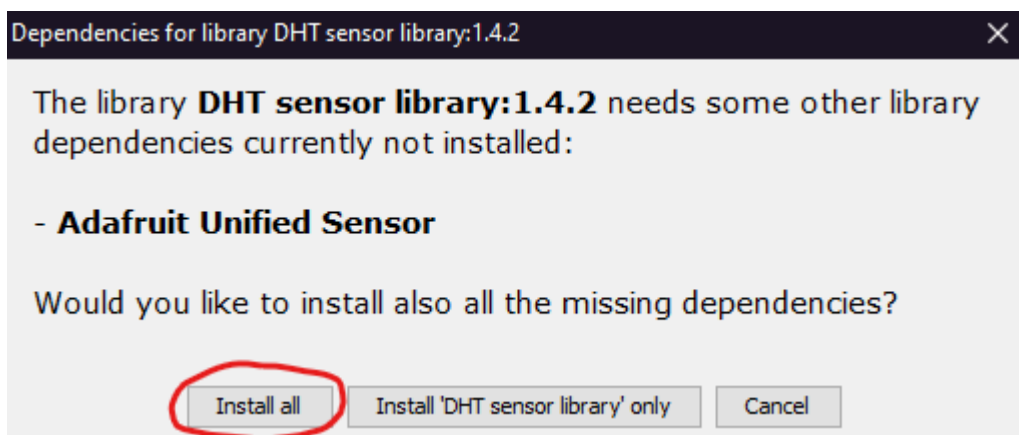


Рисунок 5.5 – Діалогове вікно з пропозицією інсталювати повний набір потрібних бібліотек.

Тепер перейдемо до самого коду. Перш за все потрібно налаштувати ESP8266 NodeMCU в режим Station (STA) і створити веб-сервер для видачі веб-сторінок будь-якому підключеному до існуючої мережі клієнту. Перш ніж приступити до завантаження скетчу потрібно щоб ESP8266 NodeMCU міг встановити з'єднання з існуючою мережею, необхідно змінити наступні дві змінні відповідно до облікових даних нашої мережі (рис.5.6).

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

Рисунок 5.6 – Поля для облікових даних мережі

Як тільки це зроблено, потрібно завантажити написаний скетч (див. нижче).

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"

#define DHTTYPE DHT22

const char* ssid = "PerseusNetwork";
const char* password = "A123456A";

ESP8266WebServer server(80);

uint8_t DHTPin = D8;

DHT dht(DHTPin, DHTTYPE);

float Temperature;
float Humidity;

void setup()
{
  Serial.begin(115200);
  delay(100);

  pinMode(DHTPin, INPUT);

  dht.begin();

  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 = dht.readTemperature();
    Humidity = dht.readHumidity();
    server.send(200, "text/html", SendHTML(Temperature,Humidity));
}

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

String SendHTML(float TempCstat,float Humiditystat)
{
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0,user-scalable=no\">\n";
    ptr += "<linkhref=\"https://fonts.googleapis.com/css?family=Open+Sans:300,400,600\"
rel=\"stylesheet\">\n";
    ptr += "<title>ESP8266 Weather Report</title>\n";
    ptr += "<style>html { font-family: 'Open Sans', sans-serif; display: block;
margin: 0px auto; text-align: center;color: #333333;}\n";
    ptr += "body{margin-top: 50px;}\n";
    ptr += "h1 {margin: 50px auto 30px;}\n";
    ptr += ".side-by-side{display: inline-block;vertical-align:
middle;position: relative;}\n";
    ptr += ".humidity-icon{background-color: #3498db;width: 30px;height:
30px;border-radius: 50%;line-height: 36px;}\n";
}

```

```

ptr += ".humidity-text{font-weight: 600;padding-left: 15px;font-size:
19px;width: 160px;text-align: left;}\n";
ptr += ".humidity{font-weight: 300;font-size: 60px;color: #3498db;}\n";
ptr += ".temperature-icon{background-color: #f39c12;width: 30px;height:
30px;border-radius: 50%;line-height: 40px;}\n";
ptr += ".temperature-text{font-weight: 600;padding-left: 15px;font-size:
19px;width: 160px;text-align: left;}\n";
ptr += ".temperature{font-weight: 300;font-size: 60px;color: #f39c12;}\n";
ptr += ".superscript{font-size: 17px;font-weight: 600;position:
absolute;right: -20px;top: 15px;}\n";
ptr += ".data{padding: 10px;}\n";
ptr += "</style>\n";
ptr += "<script>\n";
ptr += "setInterval(loadDoc,200);\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.getElementById(\"webpage\").innerHTML
=this.responseText}\n";
ptr += "};\n";
ptr += "xhttp.open(\"GET\", \"/\", true);\n";
ptr += "xhttp.send();\n";
ptr += "}\n";
ptr += "</script>\n";
ptr += "</head>\n";
ptr += "<body>\n";
ptr += "<div id=\"webpage\">\n";
ptr += "<h1>ESP8266 Weather Report</h1>\n";
ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side temperature-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_1\"
xmlns=\"http://www.w3.org/2000/svg\" xmlns:xlink=\"http://www.w3.org/1999/xlink\"
x=\"0px\" y=\"0px\" \n";
ptr += "width=\"9.915px\" height=\"22px\" viewBox=\"0 0 9.915 22\" enable-
background=\"new 0 0 9.915 22\" xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\" d=\"M3.498,0.53c0.377-0.331,0.877-
0.501,1.374-0.527C5.697-0.04,6.522,0.421,6.924,1.142\n";
ptr
+= "c0.237,0.399,0.315,0.871,0.311,1.33C7.229,5.856,7.245,9.24,7.227,12.625c1.019,
0.539,1.855,1.424,2.301,2.491\n";

```



```

ptr += "c0.491,1.163,0.518,2.514,0.062,3.693c-0.414,1.102-1.24,2.038-
2.276,2.594c-1.056,0.583-2.331,0.743-3.501,0.463\n";
ptr += "c-1.417-0.323-2.659-1.314-3.3-2.617C0.014,18.26-
0.115,17.104,0.1,16.022c0.296-1.443,1.274-2.717,2.58-3.394\n";
ptr += "c0.013-3.44,0-6.881,0.007-
10.322C2.674,1.634,2.974,0.955,3.498,0.53z\"/>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side temperature-text\">Temperature</div>\n";
ptr += "<div class=\"side-by-side temperature\">";
ptr += (int)TempCstat;
ptr += "<span class=\"superscript\">°C</span></div>\n";
ptr += "</div>\n";
ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side humidity-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_2\"
xmlns=\"http://www.w3.org/2000/svg\" xmlns:xlink=\"http://www.w3.org/1999/xlink\"
x=\"0px\" y=\"0px\" \n\"; width=\"12px\" height=\"17.955px\" viewBox=\"0 0 13
17.955\" enable-background=\"new 0 0 13 17.955\" xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\"
d=\"M1.819,6.217C3.139,4.064,6.5,0,6.5,0s3.363,4.064,4.681,6.217c1.793,2.926,2.13
3,5.05,1.571,7.057\n\";
ptr += "c-0.438,1.574-2.264,4.681-6.252,4.681c-3.988,0-5.813-3.107-6.252-
4.681C-0.313,11.267,0.026,9.143,1.819,6.217\"></path>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side humidity-text\">Humidity</div>\n";
ptr += "<div class=\"side-by-side humidity\">";
ptr += (int)Humiditystat;
ptr += "<span class=\"superscript\">%</span></div>\n";
ptr += "</div>\n";

ptr += "</div>\n";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}

```

Після завантаження скетчу потрібно відкрити монітор послідовного порту зі швидкістю 115200 біт / с і натиснути кнопку RESET на NodeMCU (рис. 5.7). Після чого він виведе динамічну IP адресу, отриману від нашої точки доступу або маршрутизатора, і покаже повідомлення про те, що HTTP сервер запущений [10].

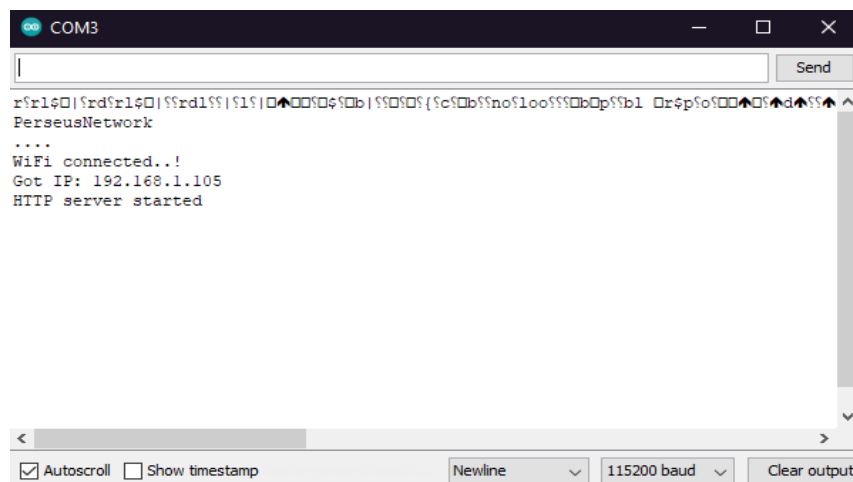


Рисунок 5.7 – Монітор послідовного порту.

Потім потрібно завантажити браузер і ввести IP адресу, вказану в моніторі послідовного порту. ESP8266 NodeMCU v3 відобразить веб-сторінку, яка показує температуру і відносну вологість (рис. 5.8).

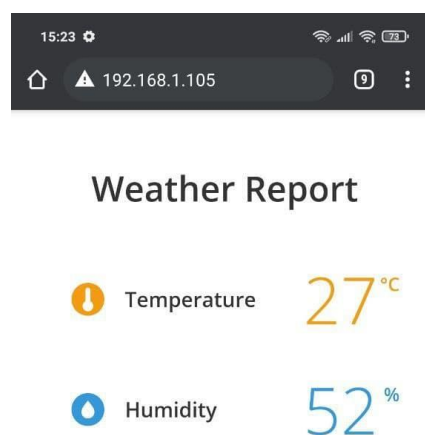


Рисунок 5.8 – Веб-сторінка із значеннями температури і вологості з датчика DHT22 на веб-сервері ESP8266.

ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи бакалавра було проведено повний огляд та аналіз аналогів елементної бази пристрою вимірювання температури і вологості повітря середовища. Сформована постановка завдань, які необхідно було виконати у ході розробки проекту.

Окрім опису аналогів було обрано програмні продукти за допомогою яких розроблявся даний програмно-апаратний комплекс. До переліку даних інструментів реалізації входять такі додатки як – Microsoft Visual Studio 2019 та Arduino IDE.

Під час створення програмно-апаратного комплексу було розроблено декілька схем. Перш за все була створена структурна схема приладу, з поясненням що до призначення кожного з наявних елементів.

Також було створено схему алгоритму функціонування програмно-апаратного комплексу, яка поетапно показує послідовність виконання інструкцій у обраної для проекту плати управління.

Після цього була сформульована та описана функціональна схема мікропроцесорного пристрою, а також розроблена принципова схема пристрою контролю у складі програмно-апаратного комплексу.

Описано процес підключення елементної бази приладу та розробки програмної частини програмно-апаратного комплексу з наявністю повного коду та інструкціями щодо програмування плати управління.

Окрім опису процесу розробки програмної частини проекту, наведені графічні матеріали етапу програмування для більш чіткого розуміння алгоритму дії під час створення подібного приладу.

					ЕліТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

СПИСОК ЛІТЕРАТУРИ

1. Ротенберг І.С., Опанасюк А.С. Пристрій для вимірювання температури і вологості повітря навколишнього середовища // Фізика, електроніка, електротехніка (ФЕЕ: 2021). Матеріали та програма міжнародної науково-технічної конференції студентів та молодих вчених (Суми, 19–23 квітня 2021 року), Суми, Сумський державний університет 2021, С.109.
2. Arduino vs ESP8266 vs ESP32 Comparison [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://diyi0t.com/technical-datasheet-microcontroller-comparison/>.
3. DHT11 vs DHT22 vs LM35 vs DS18B20 vs BME280 vs BMP180 [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/>.
4. C/C++ for Visual Studio Code [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://code.visualstudio.com/docs/languages/cpp>.
5. Arduino Software (IDE) [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://www.arduino.cc/en/guide/environment>.
6. Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>.
7. ESP8266EX Datasheet Version 4.3 [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://components101.com/asset/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf.
8. Espressif smart connectivity platform: ESP8266 [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: <https://www.electroschematics.com/wp-content/uploads/2015/02/esp8266-datasheet.pdf>.
9. Подключение датчика влажности и температуры DHT11 (DHT22) [Електронний ресурс]. – 2018. – <https://arduino-kit.ru/blogs/blog/41-podklyuchenie-datchika-vlazhnosti-i-temperature-dht11-dht22>
10. Взаимодействие ESP8266 NodeMCU с датчиками температуры и влажности DHT11 и DHT22 и вывод показаний, используя веб-сервер [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://radioprogram.ru/post/869>.

						ЕлІТ 6.171.00.10.462 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата			42