

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКИЙ ДИПЛОМНИЙ ПРОЕКТ

на тему:

Мікропроцесорна система збору даних з різнотипних датчиків

ПОЯСНЮВАЛЬНА ЗАПИСКА

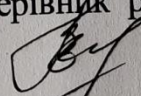
08-23.БДР.033.00.000 ПЗ

Виконав студент 2 курсу, групи 1КІ-20мс
спеціальності 123 - Комп'ютерна інженерія



Бугай Д.О.

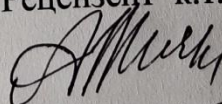
Керівник роботи к.т.н., доц. каф. ОТ



Богомолів С.В.

"18" 06 2022 р.

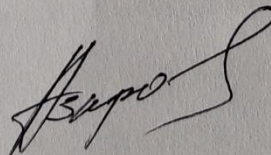
Рецензент к.т.н., доц. каф. МБІС



Шиян А.А.

"20" 06 2022 р.

Допущено до захисту
д.т.н., проф. Азаров О.Д.



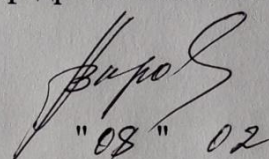
"21" 06 2022 р.

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітній рівень - магістр
Спеціальність - 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

О.Д. Азаров



"08" 02 2022 р.

ЗАВДАННЯ

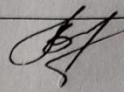
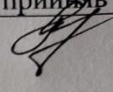
НА БАКАЛАВРСЬКУ ДИПЛОМНИЙ ПРОЕКТ

студенту Бугаю Дмитру Олександровичу

- 1 Тема проекту «Мікропроцесорна система збору даних з різнотипних датчиків», керівник проекту Богомолов Сергій Віталійович, к.т.н., доцент, затверджені наказом вищого навчального закладу від 24.03.2022 року №66
- 2 Строк подання студентом проекту 14.06.2022.
- 3 Вихідні дані до проекту: технічні параметри систем контролю складських приміщень, технічний опис ESP8266 NodeMCU та електронних компонентів, опис середовища Arduino IDE.
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд і аналіз мікропроцесорних систем збору даних, інформаційно вимірювальні системи збору даних, типи та різновиди датчиків, типи інтерфейсів, проектування мікропроцесорної системи збору даних з різнотипних датчиків, розробка схеми системи, мікропроцесорна платформа ESP8266 NodeMCU, Програмування мікропроцесорної системи, висновки, перелік джерел посилання, додатки.
- 5 Графічний матеріал (з точним зазначенням обов'язкових креслень)-схема електрична принципова.

6 Консультанти розділів роботи представлено в таблиці 1

Таблиця 1 - Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1- 3	Богомолов С.В., к.т.н., доцент кафедри ОТ		

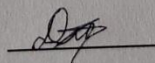
7 Дата видачі завдання 10.02.2022.

8 Календарний план виконання МКР приведений в таблиці 2.

Таблиця 2 - Календарний план

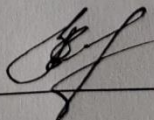
№ з/п	Назва етапів виконання бакалаврського проекту	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	14.02.22	<i>Beid</i>
2	Огляд і аналіз мікропроцесорних систем збору даних з різнотипних датчиків	15.02-28.02.22	<i>Beid</i>
3	Проектування системи	01.03-14.03.22	<i>Beid</i>
4	Реалізація програмного та апаратного забезпечення	15.03-28.03.22	<i>Beid</i>
5	Підготовка матеріалів та опис розробки	29.03-11.04.22	<i>Beid</i>
6	Оформлення пояснювальної записки та ілюстративного матеріалу	12.04-25.04.22	<i>Beid</i>
7	Аналіз виконання роботи, висновки, додатки	26.04-09.05.22	<i>Beid</i>
8	Перевірка якості виконання бакалаврського проекту та усунення недоліків	17.05.22	<i>Beid</i>

Студент



Бугай Д.О.

Керівник роботи



Богомолов С.В.

Анотація

Даний бакалаврський дипломний проект присвячений проектуванню мікропроцесорної системи контролю складських приміщень.

Пояснювальна записка складається з 74 сторінок, 35 рисунків, 4 таблиці, перелік посилань мстить 9 джерел.

В проекті виконується аналіз аналогів систем та сфери їх застосування. Виконано аналіз підходів побудови мікропроцесорних систем. Також було запроєктовано мікропроцесорну систему на базі контролера ESP8266 NodeMCU і зроблено під нього відповідне програмне забезпечення в програмному середовищі Arduino IDE.

В результаті розробки проекту розроблено мікропроцесорну систему, яка виводить інформацію про основні показники як: вологість, температура, тиск та висота.

Ключові слова: мікроконтролер, датчик, інтерфейс, алгоритм, система.

Annotation

This bachelor's thesis project is devoted to the design of a microprocessor control system for warehouses.

The explanatory note consists of 74 pages, 35 figures, 4 tables, the list of references contains 9 sources.

The project analyzes the analogues of systems and their scope. The analysis of approaches to the construction of microprocessor systems is performed. A microprocessor system based on the ESP8266 NodeMCU controller was also designed and developed in the Arduino IDE software environment.

As a result of the project development, a microprocessor system was developed, which displays information about the main indicators such as humidity, temperature, pressure and altitude.

Keywords: microcontroller, sensor, interface, algorithm, system.

ЗМІСТ

Вступ.....	7
1 Огляд та аналіз мікропроцесорної системи збору даних	9
1.1 Інформаційно вимірювальні системи збору даних	9
1.2 Типи та різновиди датчиків	11
1.3 Типи інтерфейсів Arduino 1-Wire	15
1.4 Мікропроцесорні системи	20
2 Розробка мікропроцесорної системи збору даних з різнотипних датчиків	25
2.1 Розробка структурно-функціональної схеми	25
2.2 Характеристики мікропроцесорної системи	26
2.3 Характеристики датчиків	29
3 Проектування програмної частини	34
3.1 Розробка алгоритму програми	34
3.2 Розробка програмного забезпечення для мікроконтролера	35
3.3 Розробка веб сторінки	39
3.4 Підключення приладу до локальної мережі	41
Висновки.....	44
Перелік джерел посилання	45
Додаток А Технічне завдання.....	47
Додаток Б Скетч програми та веб сторінки	50
Додаток В Структура схема плати ESP8266 NodeMCU	71
Додаток Г Повний алгоритм роботи програми	72
Додаток Д протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	74

					08-23.БДП.033.000.000 ПЗ		
<i>Розроб.</i>	<i>Бугай Д.О.</i>				Мікропроцесорна система збору даних з різнотипних датчиків Пояснювальна записка	<i>Лім.</i>	<i>Аркушів</i>
<i>Перевір.</i>	<i>Богомолов С.В</i>						74
<i>Реценз.</i>	<i>Шиян А.А.</i>					<i>ВНТУ, зр. ІКІ-20МС</i>	
<i>Н. Контр.</i>	<i>Швець С. І.</i>						
<i>Затверд.</i>	<i>Азаров О.Д.</i>						

Вступ

Збір даних - це процес отримання реального сигналу (наприклад, температура, тиск, вологість тощо) як вхідні дані в систему, для обробки, аналізу та зберігання.

Датчики використовуються для збору даних між реальним світом фізичних параметрів, який є аналоговими, та обробки мікропроцесором в цифровому середовищі, ця система збору даних призначена для збору корисних даних вимірювань визначення певних характеристик середовища, моніторингу або контролю за якимись параметрами.

Ця система може використовуватися для різноманітних операцій: системного моніторингу в сільському господарстві, військових, промислових, комерційних та для медичного застосування. Досягнення в бездротовому зв'язку, мережах і в апаратних технологіях, таких як мікроконтролери зробили революцію в розробці систем збору даних, тому тепер можна розробляти специфічні системи для додатків з розширеними можливостями для точного збору даних у реальному часі,

Тут представлені датчики для моніторингу температури, тиску та вологості навколишнього середовища. Ці три параметри навколишнього середовища впливають один на одного, і зміна одного параметру призводить до зміни інших параметрів.

Метою спроектувати мікропроцесорну систему збору даних з різнотипних датчиків, як за типом інформації, що опрацьовується, так і різними інтерфейсами.

Задачі, які необхідно виконати :

- аналіз предметної області;
- вибір платформи та компонентної бази;
- вибір засобів реалізації мікропроцесорної системи збору даних з різнотипних датчиків;
- програмування контролера ESP8266;

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		7

- тестування роботи мікропроцесорної системи;
- підключення контролера до локальної Wi-fi мережі;
- створити веб-сторінку.

Об'єктом дослідження є процеси, що відбуваються у мікропроцесорних системах опрацювання даних з різнотипних датчиків із різними інтерфейсами, наприклад, для відображення вологості і температури повітря, атмосферного тиску.

Предметом дослідження є засоби програмування мікроконтролера для реалізації мікропроцесорної системи збору даних з різнотипних датчиків.

Практичне значення полягає в тому, що розроблено Мікропроцесорна система збору даних з різнотипних датчиків, яка може застосовуватися, як у побуті, так і в різних галузях людської діяльності. Розробка є завершеною та повністю працездатною.

Апробація - це участь у ІІ науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2022 з доповіддю «Мікропроцесорна система збору даних з різнотипних датчиків». Тези опубліковано 30.05.22.

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			8

1 Огляд та аналіз мікропроцесорної системи збору даних

1.1 Інформаційно вимірювальні системи збору даних

В даний час проектуванню вимірювальних систем приділяється багато часу. Робиться великий акцент на застосування в цих системах електронно-цифрових приладів. Їхня висока швидкість вимірювання параметрів, зручна форма подання інформації, великий вибір інтерфейсів підключення, порівняно невелика похибка вимірювання порівняно з механічними та електромеханічними засобами вимірювання всі ці та багато інших переваг роблять ці типи систем перспективними для розвитку та подальшого використання в багатьох галузях промисловості.

Розвиток мікроелектроніки та широке застосування її виробів у промисловому виробництві, у пристроях та системах управління найрізноманітнішими об'єктами та процесами є нині одним із основних напрямків науково-технічного прогресу.

Використання мікроконтролерів у виробках не тільки призводить до підвищення техніко-економічних показників (надійності, потужності, габаритних розмірів), але й дозволяє скоротити час розробки виробів і робить їх уніфікованими, адаптивними, а також дозволяє зменшити їх вартість. Використання мікроконтролерів у системах управління забезпечує досягнення високих показників ефективності при низькій вартості.

Інформаційно-вимірювальні система - це сукупність вимірювальних, контрольних, діагностичних, обчислювальних, керуючих, елементів системи які реєструють, що відображається в телекомунікаційних та інших допоміжних технічних засобах , дана схема сформована для отримання вимірювальної інформації, її перетворення і обробки, передачі вимірювальної інформації за призначенням в автоматизовану систему управління або подібне .

Системи збору даних у наші дні зробили великий крок уперед і впритул наблизилися до використання досконалих електронних технологій. Зараз, багато

									Арк.А
									9
Змн.	Арк.А	№ докум.№	ПідписПі	Дата	08-23.БДП.033.00.000 ПЗ				

систем збору даних, що складаються з аналогового комутатора, підсилювача вибірки-зберігання, АЦП, стали розміщувати на одній інтегральній мікросхемі, що порівняно вплинуло на швидкість обробки даних, зручність у використанні, надійність і звичайно ж на їхню вартість.

У інформаційно вимірювальних систем є такі ознаки:

- інформаційно вимірювальні системи мають блочно-модульний принцип побудови, що дозволяє формувати систему з конструктивно завершених модулів і приладів і цей принцип забезпечує технічну та інформаційно-функціональну сумісність і взаємозамінність модулів, спрощує технічне обслуговування і підвищує точність і надійність системи;

- різномірні технічні засоби, що входять до складу системи, пов'язані єдиним алгоритмом;

- дозволяють, як правило, отримувати комплексну інформацію.

У одній системі, як правило, об'єднують технічні засоби для вимірювання, контролю різних характеристик - фізичних величин, різномірних технічних параметрів. Наприклад, інформаційно-вимірювальна система контролю якості кінопоказу вимірює комплекс параметрів кінозображення і звуковідтворення; система контролю якості телемовлення об'єднує вимір безлічі параметрів телевізійного сигналу; система контролю нанесення оптичних покриттів на оптичні компоненти елементів оптико освітлювальних систем фототехніки включає не тільки прилади для вимірювання параметрів нанесених покриттів, але і умов мікроклімату у виробничих приміщеннях і т.д. ;

Інформаційно - вимірювальні системи, як правило, не виготовляють як єдину конструкцію, вони складаються з окремих модулів, які разом з додатковими сполучними і обчислювальними елементами розміщують в контрольованому середовищі;

Структура інформаційно-вимірювальної системи, як правило, передбачає централізоване управління окремими елементами і багатоканальні вимірювання різних параметрів.

									Арк.А
									10
Змн.	Арк.А	№ докум.№	ПідписПі	Дата					

08-23.БДП.033.00.000 ПЗ

Прикладами інформаційно вимірювальних систем, що використовуються в приладобудуванні, також є:

- система для оцінки і дослідження шорховатості поверхностей деталей фототехніки, що включає профілограф-профілометр з цифровим відліковим пристроєм, модулем, що забезпечує реєстрацію аналізованих профілограм, додаткове приладдя для розміщення і закріплення контрольованих деталей різної конфігурації;

- мультисенсорна вимірювальна машина (системи вимірювань лінійних і кутових величин в полярних і прямокутних координатах з автоматичною обробкою результатів вимірювань, відображенням і реєстрацією отриманих результатів; це стандартні інформаційно вимірювальні системи, що випускаються підприємствами з виробництва вимірювальної техніки;

- блочно модульні комплекси, призначені для радіовимірювання;

- системи, що забезпечують передачу вимірювальних сигналів між окремими компонентами системи і виконання різних обчислювальних операцій.

1.2 Типи та різновиди датчиків

В сучасному світі датчики займають таку саму високу позицію як і мікропроцесори, Ви можете знайти різнотипні датчики в наших будинках, офісах, автомобілях та електроніці які працюють для того щоб зробити наше життя комфортніше.

Вони вмикають світло, помічають нашу присутність, регулюють температуру та вологість в приміщенні, виявляють задимленість або вогонь, готують нам смачну каву , відкривають гаражні ворота як тільки наша машина наближується до них, та виконують багато інших задач.

Всі ці задачі автоматизації можливі завдяки датчикам. Перед тим як перейти до подальшого опису того що таке датчик, які є типи датчиків та області їх використання⁷, розглянемо приклад автоматизованої системи яка можлива завдяки датчикам.

									Арк.А
									11
Змн.	Арк.А	№ докум.№	ПідписПі	Дата	08-23.БДП.033.00.000 ПЗ				

Розглянемо як приклад таку систему як автопілот в літаках Майже всі цивільні та воєнні літаки мають таку функцію як автоматична система управління польотом , а саме автопілот , він складається з датчиків які використовуються для різноманітних задач , таких як контроль швидкості , гіроскоп , контроль висоти , gps навігація , герметичність , рівень палива та багато іншого. Потім центральний комп'ютер обробляє дані з усіх датчиків та зрівнює їх з еталонними значеннями , після цього комп'ютер передає сигнали управління на різні модулі літака такі як двигуни закриття та елерони що дає можливість зробити політ плавним.

Датчик це пристрій який перетворює фізичний показник (температуру , вологість , положення в просторі) в електричний сигнал.

Найпростіший приклад датчика це фоторезистор. Це сенсор опір якого змінюється в залежності від інтенсивності освітлення. Ми можемо підключити його до дільника напруги і перевірити падіння напруги на фоторезисторі , цю напругу можна відкалібрувати. Таким чином у нас вийде датчик освітлення.

Датчики поділяються на цифрові та аналогові. Аналогові дають аналоговий вихідний сигнал. Цифрові датчики на відміну від аналогових працюють з дискретними або цифровими даними. Данні в цифрових датчиках які використовуються для передачі та перетворення являються цифровими по своїй природі

Знизу приведемо список різних типів датчиків:

- датчик температури;
- датчик наближення;
- акселерометр;
- інфрачервоний датчик;
- датчик світла;
- ультразвуковий датчик;
- датчик диму;
- датчик алкоголю;
- сенсорний датчик;

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		12

Ультразвуковий датчик - це пристрій для вимірювання відстані та швидкості об'єкта. Даний датчик працює на основі властивостей звукових хвиль із частотою, що перевищує чутний людиною діапазон. Використовуючи час проходження звукової хвилі, він визначає відстань до об'єкта. А для вимірювання швидкості використовується властивість доплерівського зрушення.



Рисунок 1.3 - Makeblock me ultrasonic sensor v3

Датчик світла, також відомі як фотодатчики, датчики світла є одним із найважливіших датчиків. Простим світловим датчиком, доступним сьогодні, є фотозалежний резистор або LDR. Властивість LDR полягає в тому, що його опір обернено пропорційний інтенсивності навколишнього освітлення, коли інтенсивність світла збільшується, його опір зменшується і навпаки.

Використовуючи схему LDR, ми можемо відкалібрувати зміни її опору для виміру інтенсивності світла. Також існують два інших типи датчиків світла (або фотодатчики), які часто використовують в складних електронних схемах. Це фотодіод та фототранзистор. Все це аналогові датчики.

Існують також цифрові фотодатчики, такі як BH1750, TSL2561 тощо, які можуть обчислювати інтенсивність світла та надавати цифрове еквівалентне значення.



Рисунок 1.4 - фотодатчик BH1750

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		14

Датчики вологості - якщо ви бачите системи моніторингу погоди, вони постійно надають дані про температуру та вологість. З цього випливає що вимірювання вологості є важливим завданням у багатьох галузях, і датчики вологості допомагають нам це досягти. В основному датчики вологості вимірюють відносну вологість (відношення вмісту води в повітрі до максимальної можливості повітря утримувати воду). Оскільки відносна вологість залежить від температури повітря, майже всі датчики вологості також можуть вимірювати температуру.

Датчики вологості поділяються на ємнісні, резистивні та теплопровідні.



Рисунок 1.5 - датчик вологості DHT22

1.3 Типи інтерфейсів Arduino 1-Wire

UART, I2C і SPI є одними з найпоширеніших і основних апаратних комунікаційних периферійних пристроїв, які виробники та електрики використовують при розробці мікроконтролерів. Аналогічно, для Arduino вони також містять периферійні пристрої UART, I2C і SPI.

UART порт Arduino - це універсальний асинхронний порт прийому та передачі який являє собою протокол послідовного зв'язку який дозволяє Arduino взаємодіяти з послідовними пристроями. UART може підтримувати двонаправлену, асинхронну та послідовну передачу даних. Також він присутній на всіх типах плат Arduino що дозволяє їм обмінюватися даними з ПК завдяки інтегрованому перетворювачу USB в послідовний порт. Якщо програма написана для ОС Windows, Mac або Linux і ви хочете використовувати цю програму за

									Арк.А
									15
Змн.	Арк.А	№ докум.№	ПідписПі	Дата					

08-23.БДП.033.00.000 ПЗ

допомогою Arduino, то можете просто просто об'єднати їх разом через USB-порт, якщо це був послідовний порт.

Переваги використання UART з Arduino:

- простота в експлуатації та використанні з Arduino, він добре задокументований на багатьох ресурсах, так як він широко використовується користувачами Arduino;

- годинник не потрібен.

Недоліки використання UART з Arduino:

- більш низька швидкість відповідно до I2C і SPI;

- швидкість передачі кожного UART повинна бути в межах 10% від іншого, щоб запобігти втраті даних;

- недопустиме використання декількох провідних систем, таких як (Arduino and slaves).

Figure 295. IrDA data modulation (3/16) -normal mode

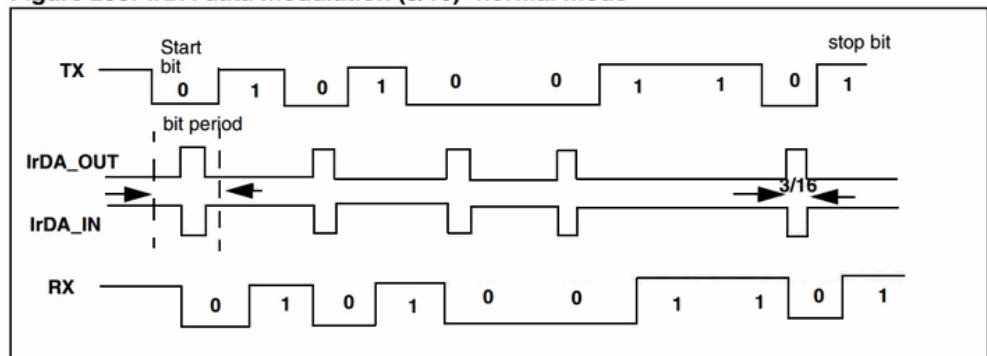


Рисунок 1.6 - протокол роботи UART

I2C - розшифровується як інтерфейс послідовного зв'язку, розроблений для мікроконтролерів. Він схожий на UART але не використовується для зв'язку з ПК та датчиками. Він являє собою двонаправлену синхронну послідовну шину з двома контактами для передачі інформації підключеними до шини.

I2C корисний для проектів Arduino, оскільки вони іноді вимагають спільної роботи багатьох різних частин (наприклад, датчиків, розширення,

драйверів), а з I2C ви можете підключити до 128 пристроїв на материнській платі, зберігаючи при цьому чіткий шлях зв'язку між основною платою та датчиками або іншими пристроями.

Переваги використання I2C з Arduino:

- низька кількість контактів/сигналів навіть для багатьох пристроїв на шині
- підтримує комунікацію з кількома провідними та кількома відомими (наприклад, можна підключити Raspberry Pi разом з Arduino);
- використовує лише 2 дроти;
- адаптація до потреб різних підпорядкованих пристроїв.

Недоліки використання I2C з Arduino:

- нижча швидкість порівняно з SPI (наступним периферійним пристроєм зв'язку), оскільки він використовує підтягуючі резистори;
- конструкція з відкритим стоком обмежена швидкість;
- потребує більше місця за рахунок використання резисторів;
- може стати складним із збільшенням кількості пристроїв.

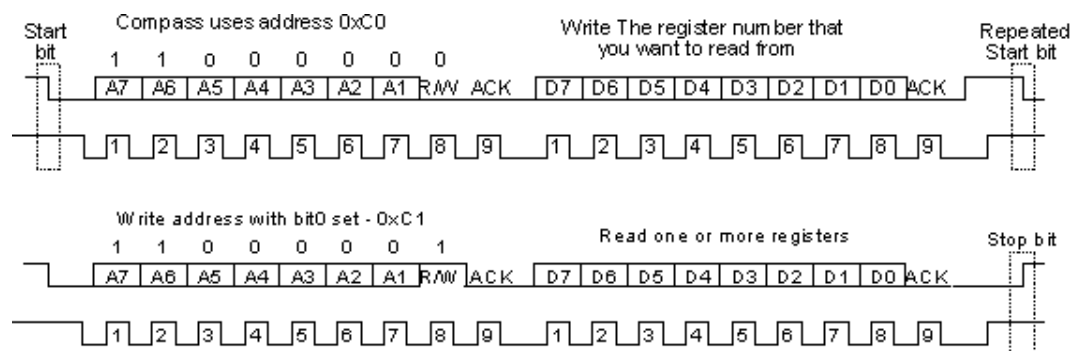


Рисунок 1.7 – протокол роботи I2C

SPI порт - це послідовний периферійний інтерфейс схожий на I2C

але з іншою формою протоколу послідовного зв'язку, спеціально розробленому для підключення мікроконтролерів. SPI працює в повному дуплексі де дані можна надсилати та отримувати одночасно.

У порівнянні з UART і I2C, в цього порта пропускна здатність 8 Мбіт або

більше. Зазвичай ця швидкість виникає завдяки своєму простому протоколу. Лінії даних/годинника є спільними між пристроями, і кожен пристрій потребує унікального адресного контакту SPI використовується в місцях, де швидкість важлива, як-от SD-карти, модулі та дисплеї або коли інформація швидко оновлюється та змінюється, як в термометрах. Однак SPI можна використовувати лише з одним головним пристроєм, яким є Arduino та 4 підлеглими пристроями максимально .

Переваги та недоліки використання SPI з Arduino:

- а) переваги використання SPI з Arduino;
 - протокол простий, оскільки не існує складної підпорядкованої системи адресації, як-от I2C;
 - найшвидший протокол порівняно з UART та I2C;
 - немає стартових і стопових бітів, на відміну від UART, що означає, що дані можуть передаватися безперервно;
- б) недоліки використання SPI з Arduino;
 - більше Pin-портів зайнято, практичне обмеження для кількості пристроїв. (1 головний і 4 підпорядкованих пристрої);
 - не вказано керування потоком, а механізм підтвердження не підтверджує, чи отримані дані, на відміну від I2C;
 - використовує чотири лінії – MOSI, MISO, NCLK, NSS;
 - немає форми перевірки помилок, на відміну від UART.

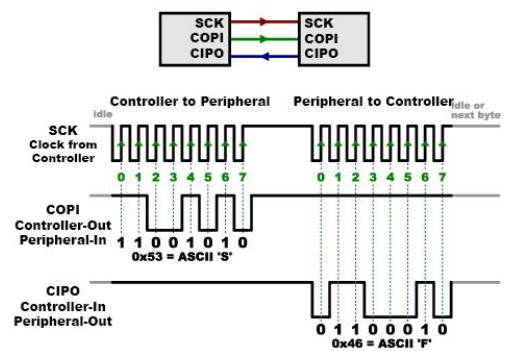


Рисунок 1.8 – протокол роботи SPI

Однопровідний інтерфейс 1-wire, представляє собою інформаційну мережу яка використовується для цифрового зв'язку за допомогою однієї лінії. Таким чином для реалізації середовища обміну цієї мережі може бути використана неекранована вита пара або навіть звичайний телефонний провід.

Переваги використання 1-wire з Arduino:

- простий протокол;
- проста структура зв'язку;
- низьке енергоспоживання;
- виключна дешевизна технології в цілому;
- датчик можна підключити на великій відстані;
- легка зміна конфігурації.

- недоліки 1-wire з Arduino:

- підтримує лише низьку швидкість передачі даних;
- пристрої 1-wire виготовляються лише Dallas Semiconductor;
- інтерфейс реалізований як апаратно так і програмно тому синхронізація даних відбувається за допомогою програмного забезпечення що являє собою складну задачу;

- погана завадостійкість на великій відстані.

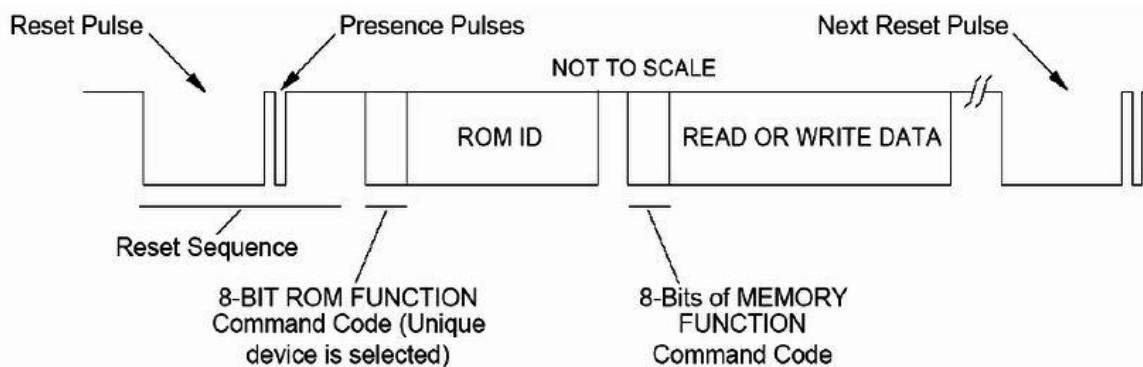


Рисунок 1.9 – протокол роботи 1-wire

1.4 Мікропроцесорні системи

Мікропроцесорні системи призначені для автоматизації обробки інформації і управління різними процесами. Мікропроцесорна система включає в себе апаратне забезпечення (hardware) і програмне забезпечення (software). Основу мікропроцесорної системи становить мікропроцесор (процесор), який виконує функції обробки інформації та управління. Решта пристроїв, що входять до складу мікропроцесорної системи, обслуговують процесор, допомагаючи йому працювати. Обов'язковими пристроями для створення мікропроцесорної системи є порти вводу/виводу і частково пам'ять. Порти вводу/виводу пов'язують процесор з зовнішнім світом, забезпечуючи вводу інформації для обробки і виведення результатів обробки, або керуючих дій. До портів введення підключають кнопки (клавіатуру), різні датчики; до портів виведення – пристрої, які допускають електричне управління: індикатори, дисплеї, контактори, електроклапани, електродвигуни і т.д. Пам'ять потрібна в першу чергу для зберігання програми (або набору програм), необхідної для роботи процесора. Програма – це послідовність команд, зрозумілих процесору, написана людиною (програмістом). Мікропроцесорні системи поділяються на універсальні, спеціалізовані й багатопроцесорні.

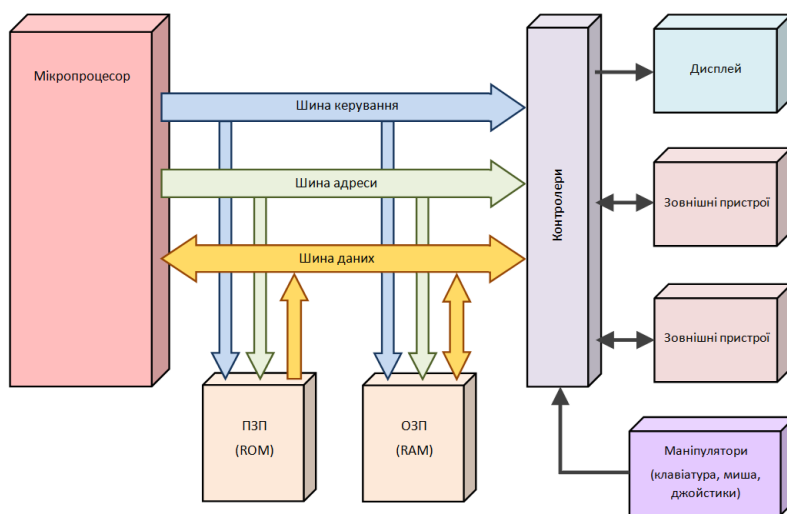


Рисунок 1.10 – класична мікропроцесорна система

Змн.	Арк.А	№ докум.№	ПідписПі	Дата

Універсальні МПС орієнтовані на розв'язання широкого класу задач:

- мікро-ЕОМ – комп'ютер малих розмірів на основі мікропроцесора. Це найбільш поширений клас МПС для різних технічних систем;

- ПЕОМ – персональні мікро-ЕОМ, розраховані на масового користувача; - робочі станції – різні автоматизовані робочі місця; - сервери – виконують частину функцій з обслуговування груп користувачів(розподіл ресурсів пам'яті, баз даних, принтерів та ін.);

- мейнфрейми – великі універсальні комп'ютери;

- кластерні системи – об'єднання машин з єдиним механізмом керування й програмного забезпечення, які забезпечують розподіл ресурсів, високу готовність, зручність розширення конфігурації.

Спеціалізовані МПС орієнтовані на вирішенню спеціалізованих завдань керування чи обробки інформації у складі технічних систем, цифрової обробки сигналів, обробки графіки.

Багатопроцесорні системи забезпечують функціонування багатьох процесорів під спільним керування. Основними технічними характеристиками МПС є: розрядність, ємність пам'яті, продуктивність, кількість зовнішніх пристроїв та їхня пропускна здатність, функції системи и склад програмного забезпечення.

Розрізняють такі основні типи МПС:

- мікроконтролери являють собою найбільш простий тип, в якому усі або більшість вузлів системи представлені у вигляді однієї мікросхеми;

- контролери є керуючими МПС, виконаними у вигляді окремих модулів; - мікрокомп'ютери – потужніші МПС, що мають розвинені засоби сполучення з зовнішніми пристроями; комп'ютери будь-якого типу є найпотужніші і найбільш універсальні МПС.

Чітку межу між названими типами іноді провести досить складно, оскільки швидкодія всіх типів мікропроцесорів постійно зростає, і буває, що новий

									Арк.А
									21
Змн.	Арк.А	№ докум.№	ПідписПі	Дата	08-23.БДП.033.00.000 ПЗ				

мікроконтролер може виявитися швидшим ніж, наприклад, застарілий комп'ютер. Але принципи відмінності між ними все ж існують.

Мікроконтролери є універсальними пристроями, практично завжди використовуються в складі більш складних пристроїв, в тому числі і контролерів. Системна шина мікроконтролера знаходиться всередині мікросхеми. Можливості підключення зовнішніх пристроїв до мікроконтролера обмежені. Пристрої, побудовані на мікроконтролерах, як правило, використовуються для виконання одного завдання.

Мікропроцесор являє собою програмно керований пристрій призначено для обробки цифрової інформації та управління процесом цієї обробки та виконано у вигляді однієї або кількох великих інтегральних схем . Мікропроцесори характеризуються великою кількістю параметрів та властивостей, оскільки вони є, з одного боку, функціонально складним обчислювальними пристроями, а з другого електронним приладом, виробом електронної промисловості. Як засіб обчислювальної техніки він характеризується в першу чергу своєю архітектурою, тобто сукупністю програмно-апаратних характеристик, що надаються користувачеві. Сюди відносяться система команд, типи та формати даних, що обробляються, режими адресації, кількість та розподіл регістрів, принципи взаємодії з оперативною пам'яттю та зовнішніми пристроями.

Контролери зазвичай створюють для вирішення окремого завдання або групи близьких задач. Вони не мають можливості підключення додаткових вузлів і пристроїв (великий пам'яті, засобів вводу/виводу). Їх системна шина, як правило, недоступна для користувача. За структурою контролер простий і оптимізований під максимальну швидкодію. В основному виконувани ним програми зберігаються в постійній пам'яті і не змінюються. Конструктивно контролери випускаються у вигляді однієї плати.

Мікрокомп'ютери відрізняє від контролерів відкритіша структура, оскільки в них допускається підключення до системної шини кількох додаткових пристроїв. Випускаються мікрокомп'ютери в каркасі, корпусі з роз'ємами

										Арк.А
										22
Змн.	Арк.А	№ докум.№	ПідписПі	Дата	08-23.БДП.033.00.000 ПЗ					

системної магістралі, які доступні для користувача. Мікрокомп'ютери мають засоби зберігання інформації на магнітних носіях (магнітні диски) і розвинені засоби зв'язку з користувачем (монітор, клавіатуру). Мікрокомп'ютери призначені для вирішення ширшого кола завдань, ніж контролери, проте до кожної нової задачі їх потрібно пристосовувати заново. Програми, що виконуються мікрокомп'ютером, можна легко замінювати.

Комп'ютери, в тому числі і персональні, являють собою найбільш гнучкі МПС. У них передбачена можливість удосконалення, а також широкі можливості підключення нових пристроїв. Системна шина комп'ютерів є доступною для користувача. Крім цього зовнішні пристрої мають можливість підключення до комп'ютера через кілька вбудованих портів зв'язку (кількість портів може доходити до 10). Комп'ютер має високо розвинені засоби зв'язку з користувачем, засоби тривалого зберігання інформації великого обсягу, засоби зв'язку з іншими комп'ютерами через інформаційні мережі. Області застосування комп'ютерів найрізноманітніші: від математичних розрахунків і обслуговування доступу до баз даних до управління роботою складних електронних систем, комп'ютерних ігор тощо.

Процесором (П) називається пристрій з програмним управлінням, що призначений для алгоритмічної обробки цифрової інформації. Процесор, реалізований у вигляді великої (ВІС) або надвеликої інтегральної мікросхеми (НВІС), називають мікропроцесором (МП). Мікропроцесор відіграє важливу роль в цифрових системах різного призначення. Це системи обробки інформації, управління об'єктами і процесами, інформаційно-вимірювальні системи та інші, що застосовуються у промисловості та різних інших галузях.

За функціональними ознаками, що обумовлюють специфічні варіанти застосування в окремій області, можна виділити наступні класи мікропроцесорів: мікропроцесори загального призначення (універсальні мікропроцесори), проблемна орієнтовані та спеціалізовані мікропроцесори. Серед проблемна орієнтованих мікропроцесорів найбільш широке розповсюдження отримали

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			23

мікроконтролери та цифрові сигнальні процесори.

До спеціалізованих належать мікропроцесори, що виготовлені для використання тільки в певних пристроях, наприклад, в пристроях мобільного зв'язку, медійних приладах і таке інше. Мікропроцесори загального призначення використовуються для вирішення широкого кола задач обробки різноманітної інформації і знаходить застосування в персональних комп'ютерах, робочих станціях, серверах і інших цифрових системах масового застосування. Зазвичай це 32-, 64- та 120-розрядні мікропроцесори. У більшості випадків вони мають суперскалярну структуру, коли декілька операційних пристроїв виконують одночасну обробку даних.

До найбільш відомих розробників мікропроцесорів цього класу належать компанії INTEL, AMD (Advanced Micro Devise), MOTOROLA, SUN MICROSYSTEMS та інші. Мікроконтролери орієнтовані на рішення задач управління (control - управління). Мікроконтролери застосовуються для побудови приладів управління для різноманітних систем.

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			24

2 Розробка мікропроцесорної системи збору даних з різномісних датчиків

2.1 Розробка структурно-функціональної схеми

В основі моєї бакалаврської роботи покладена реалізація взаємодії між мікроконтролером та датчиками які передають інформацію мікроконтролеру для подальшого її використання.

Для цього використовуються різномісні датчики, а саме такі: датчик температури, датчик вологості і датчик атмосферного тиску. Тобто з датчика атмосферного тиску ми отримуємо тиск у міліметрах ртутного стовпчика та висоту яку в свою чергу ми обраховуємо із температури та тиску.

Розробка принципової схеми даного пристрою надасть можливість продемонструвати набір і роботу необхідних нам елементів. Структурна схема зображена на рисунку 2.1

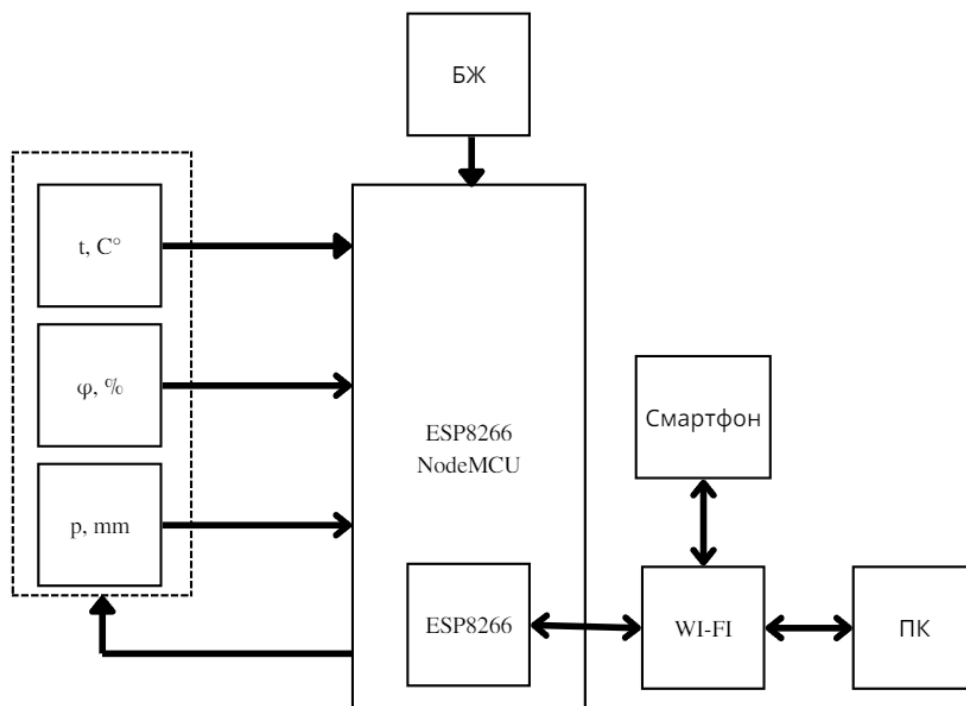


Рисунок 2.1 - Структурно-функціональна схема пристрою

Головним компонентом мого проекту є мікроконтролер ESP8266 NodeMcu який в свою чергу комутує з блоком датчиків, до якого входять: датчик

Змн.	Арк.А	№ докум.№	ПідписПі	Дата

температури, датчик вологості та датчик тиску.

Але для функціонування пристрою потрібно підключити наступні елементи. Блок живлення (БЖ) напевно найголовніший елемент інтегрований в головну плату, який живить саму плату та все що до неї підключено.

Блок датчиків являє собою комплекс датчиків який збирає інформацію з навколишнього середовища.

ESP8266 представляє собою, WI-FI модуль для збору або передачі інформації.

Роутер (WI-FI) пристрій який надає змогу комутувати пристрою з сторонніми приладами такими як ПК, ноутбуки та смартфони.

2.2 Характеристики мікропроцесорної системи

Відповідно до мого технічного завдання цей пристрій виготовлений на основі відлагодженої макетної плати типу ESP8266 модель (ESP-12E NODEMCU).

NodeMCU ESP8266 – це контролер з відкритим кодом на основі Lua та плати розробки яка спеціально призначена для програм та додатків на основі інтернет речей. Він включає в себе прошивку, що працює на SoC ESP8266 Wi-Fi від Espressif Systems, і апаратне забезпечення, розроблене для модуля ESP-12, і, таким чином, його також можна запрограмувати за допомогою середовища розробки Arduino IDE. Також ця плата може діяти як точка доступу Wi-Fi або може підключатися до неї. Вона має один контактний аналоговий вхід, 16 контактів цифрового введення-виведення, а також можливість підключення до протоколів послідовного зв'язку, таким як SPI, UART та I2C. NodeMCU має 128 КБ ОЗУ та 4 МБ флеш-пам'яті для зберігання даних та програм. Його висока обчислювальна здатність із вбудованими функціями Wi-Fi/Bluetooth і режим глибокого сну робить його ідеальним для проектів інтернет речей. Його додатки включають прототипи пристроїв інтернет речей, простих модулів з акумуляторним живленням і проектами для яких потрібні інтерфейси введення-виведення з можливостями Bluetooth та Wi-Fi.

										Арк.А
										26
Змн.	Арк.А	№ докум.№	ПідписПі	Дата	08-23.БДП.033.00.000 ПЗ					

ESP8266 містить вбудований репітер Wi-Fi 802.11b/g/n HT40, тому він може не тільки підключатися до мережі Wi-Fi і взаємодіяти з інтернетом, але і встановлювати власну мережу, дозволяючи іншим пристроям підключатися безпосередньо до нього. Це робить ESP8266 NodeMCU ще більш універсальним.

NodeMCU може житися від роз'єму Micro USB та контакту VIN (контакт зовнішнього живлення). Він підтримує інтерфейс UART, SPI та I2C.

Як на платах Arduino на NodeMcu встановлено світлодіод, який підключений до виводу D0 (GPIO2). Так само на платі передбачені дві кнопки, перша підписана як RST і розміщена у верхньому лівому кутку, є кнопкою скидання, яка використовується для скидання мікросхеми ESP8266. Друга кнопка FLASH у лівому нижньому куті, кнопка завантаження, що використовується при оновленні прошивки, виведення GPIO0.

Таблиця 2.1 – Характеристики мікроконтролера ESP 8266

Мікроконтролер	ESP-12E
Робоча напруга	3.3В
Напруга живлення	Від 3.3 до 6В
Напруга живлення (гранична)	Від 3.7 до 20В
Цифрові входи/виходи	17
Аналогові входи	2
Струм одного входу/виходу	300 мА
Струм на контакт загального значення	6 мА
Flash-пам'ять	4 МВ
SRAM	Від 64 КБ
Тактова частота	2.4 МГц
RAM	4 МВ

Розпіновка плати NodeMCU ESP8266 зображена на рисунку 2.1.

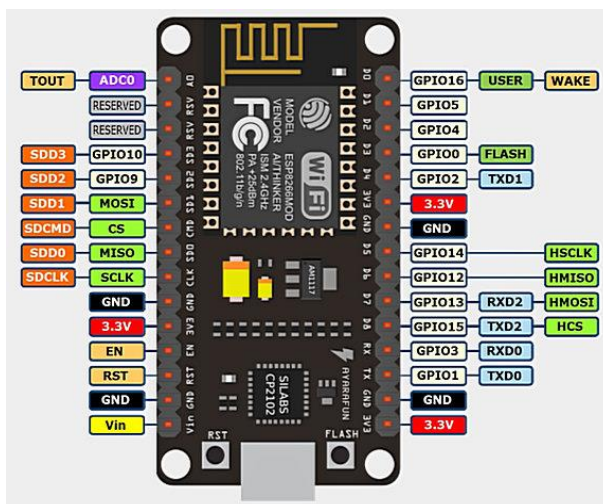


Рисунок 2.2 - Структура плати NodeMCU ESP8266

Призначення контактів:

- вивід VIN вивід для підключення зовнішнього живлення;
- вивід 3V3 і GND вихід з внутрішнього стабілізатора напруги, можна використовувати для подачі живлення на зовнішній пристрій;
- RST вивід використовується для скидання мікросхеми ESP8266;
- EN коли на виході висока напруга мікросхема працює, коли низька, мікросхема переходить в режим енергозбереження ;
- CLK (GPIO6), SDO (GPIO7), CMD (GPIO11), SD1 (GPIO8), SD2 (GPIO9) и SD1 (GPIO10) виходи підключені до Flash чіпу в модулі ESP-12E не рекомендується використовувати;
- PSW зарезервовані контакти;
- A0 вивід вбудованого 10-розрядного АЦП;
- TX (GPIO1), RX (GPIO3) контакти для завантаження прошивки і зв'язку з комп'ютером;

Змн.	Арк.А	№ докум.№	ПідписПі	Дата

08-23.БДП.033.00.000 ПЗ

Арк.А

28

- D0 (GPIO16), D1 (GPIO5), D2 (GPIO4), D3 (GPIO0), D4 (GPIO2), D5 (GPIO14), D6 (GPIO15), D7 (GPIO13), D8 (GPIO15) виводи для підключення зовнішніх пристроїв.

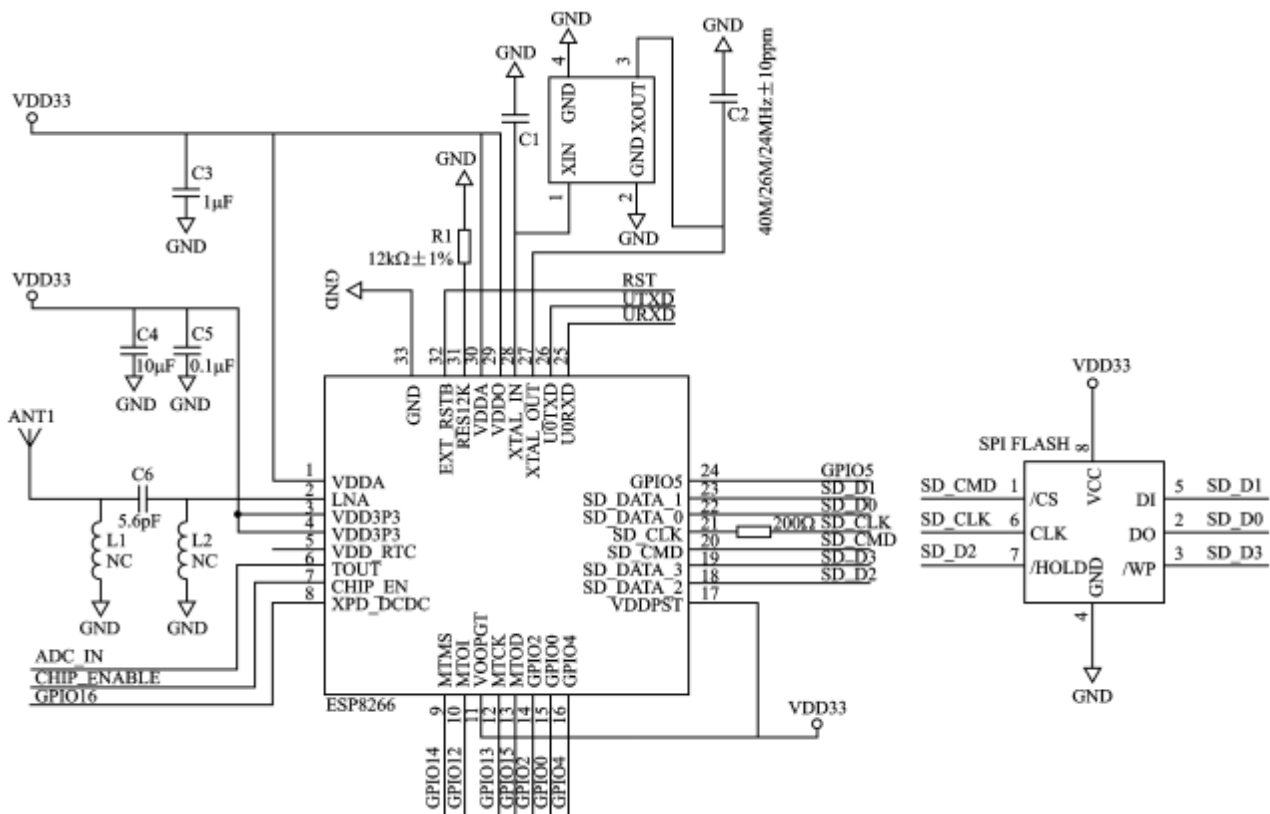


Рисунок 2.3– Принципова схема NodeMCU ESP8266

2.3 Характеристики датчиків

В даній схемі використовуються такі датчики як : BMP280, DS18B20, DHT11.

BMP280 – високоточний датчик атмосферного тиску на базі чіпа BOSH. Даний датчик калібрується у заводському середовищі. Його малі розміри та низьке енергоспоживання разом з точністю вимірювальних даних за що його люблять розробники Arduino-проектів. Модифікація цього датчика надає користувачеві 2 послідовні інтерфейси обміну даними (SPI та I2C).

У датчика є 3 режима роботи:

Змн.	Арк.А	№ докум.№	ПідписПі	Дата
------	-------	-----------	----------	------

- NORMAL – в даному режимі модуль прокидається з певною періодичністю, виконує необхідні вимірювання та знову засинає, частота вимірювань задається програмним шляхом, а результат записується або зчитується, при необхідності;

- SLEEP – режим максимально зниженого споживання енергії;

- FORCED – цей режим дозволяє будити модуль подачею зовнішнього сигналу керування; після виконання вимірювань модуль автоматично переходить в режим зниженого енергоспоживання.

Існує кілька напрямків застосування модулів BMP280. Хтось використовує їх у складі польотних контролерів для визначення висоти або як глибиномір, наприклад при зануренні в шахту. Але основним напрямком є збирання даних для метеостанцій. Для більш тісного знайомства з модулем, створимо свій проект домашньої метеостанції з виведенням погодної інформації на графічний РК-дисплей від NOKIA 5110. Щоб зробити проект цікавішим, внизу буде виводитись графік зміни атмосферного тиску. Такий підхід дозволить спрогнозувати наближення дощу з різкого падіння тиску або хорошу погоду щодо його динамічного зростання. Управлятиме всім цим плата Arduino Nano. На малюнку №5 наведено схему проекту домашньої метеостанції.

Таблиця 2.3 - Характеристики датчика BMP280

Напруга живлення	Від 1.71 В до 3.6 В
Інтерфейс обміну даними	I2C SPI
Струм споживання	0.27 А
Діапазон виміру атмосферного тиску	Від 300hPa до 1100hPa
Діапазон виміру температури	Від мінус 40°C до 85°C
Максимальна частота роботи інтерфейсу I2C	3.4 МГц;
Максимальна частота роботи інтерфейсу SPI	10 МГц;

DC5V – цифровий датчик температури повітря розроблений на мікросхемі DS18D20, яка перетворює температуру корпусу сенсора в дані, що передаються по послідовній двопровідній шині даних 1-Wire. Цей модуль призначений для роботи спільно з мікроконтролерами, зокрема на базі Arduino

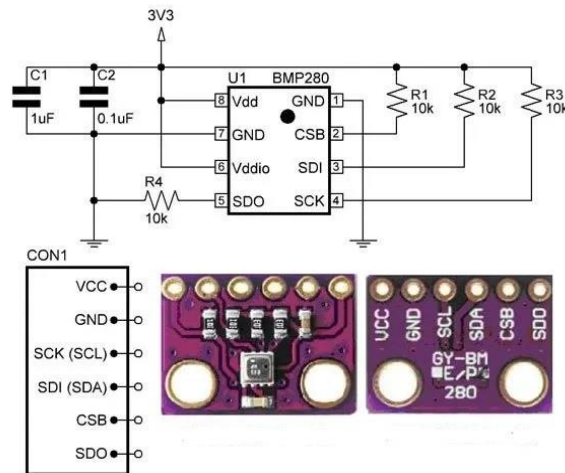


Рисунок 2.4 - Зовнішній вигляд та принципова схема модуля BMP280

Модуль датчика температури DS18B20 застосовується для вимірювання температури повітря як у приміщенні, так і на відкритому повітрі. За допомогою цього датчика можна збирати кімнатні термометри та автоматичні метеостанції. DS18B20 часто використовується при розробці системи "Розумний дім".

Внутрішня енергонезалежна пам'ять температурних установок забезпечує запис довільних значень верхньої та нижньої межі установок. Крім того, мікросхема містить вбудований логічний механізм пріоритетної сигналізації в лінію про факт виходу температури за один з обраних порогів.

Термометр має індивідуальний 64-розрядний реєстраційний номер (груповий код 028H) і забезпечує можливість роботи без зовнішнього джерела живлення, тільки за рахунок паразитного живлення однопровідної лінії. Живлення приладу через окремий зовнішній вивід здійснюється напругою від 3.0В до 5.5В.

Таблиця 2.3 – Характеристики датчика DS18B20

Напруга живлення	Від 3.5В до 5 В
Струм живлення	0,10 А
Діапазон виміру температури	Від мінус 55 до 125°
Роздільна здатність	Від 9 біт до 12 біт
Точність виміру	0.5°
Час виміру	0.75 с

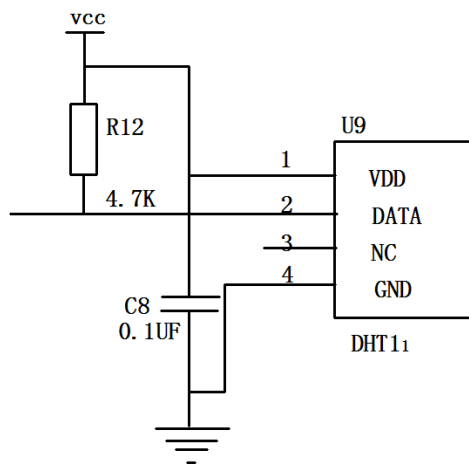


Рисунок 2.5 – Принципова схема модуля DS18B20



Рисунок 2.6 - Зовнішній вигляд та схема модуля DS18B20

DHT11 - це цифровий датчик вологості та температури, який складається з термістора та ємкісного датчика вологості. Також датчик містить у собі АЦП для перетворення аналогових значень вологості та температури. Датчик DHT11 не проявляє високу швидкодію та точність, проте простий, бюджетний і відмінно підходить для навчання і контролю вологості в приміщеннях.

При підключенні до мікроконтролера, ви можете між виводами Vcc і Data розмістити підтягуючий pull-up резистор номіналом 10 кОм. Плата Arduino має вбудовані pull-up, однак вони дуже слабенькі - близько 100 кОм.

Таблиця 2.4 - Характеристики датчика DHT11

Живлення	Від 3.5 В до 5.5 В
Струм живлення	В режимі вимірювання 0.3А в режимі очікування 0.006 А
Визначення вологості	20-80% з точністю в 5%
Визначення температури	0-50° з точністю в 2%
Частота опитування	1 гц

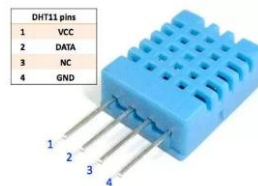


Рисунок 2.7 - Зовнішній вигляд та схема модуля DHT11

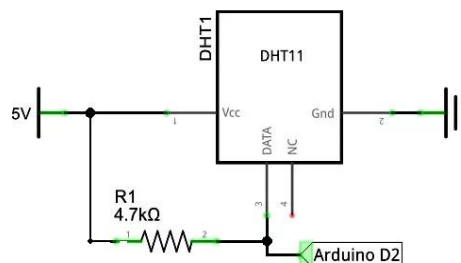


Рисунок 2.8 - Принципова схема DHT11

3 Проектування програмної частини

3.1 Розробка алгоритму програми

Робота будь-якого пристрою на основі мікроконтролера потребує прошивки, яку можна розробити в певному середовищі програмування, в даному випадку Arduino IDE.

Алгоритм - це процедура, яка використовується для розв'язування задачі або виконання обчислень. Алгоритми діють як точний список інструкцій, які виконують певні дії крок за кроком у програмних або апаратних програмах.

Всі мікропроцесорні системи потребують для роботи програмне забезпечення. На рисунку 3.1 зображена блок-схема для роботи програми.

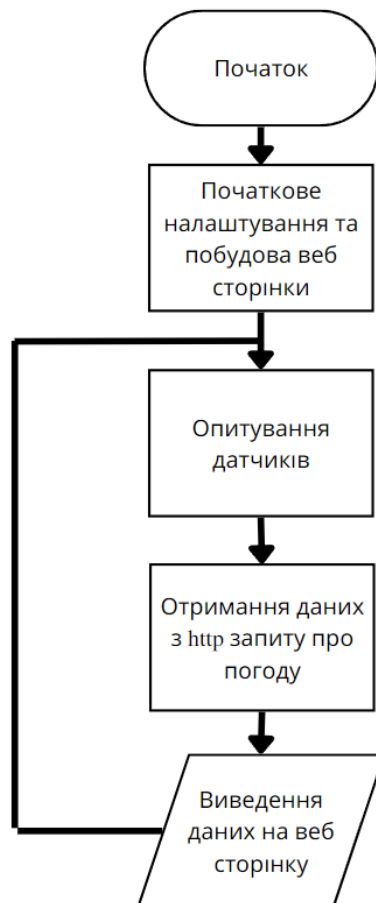


Рисунок 3.1 - Блок схема алгоритму системи

Змн.	Арк.А	№ докум.№	ГідписПі	Дата

Алгоритм можна описати так:

- крок 1 - початкове налаштування приладу та побудова веб сторінки;
- крок 2 - опитування всіх підключених датчиків;
- крок 3 - отримання інформації про погоду;
- крок 4 - запис показників датчиків в буфер;
- крок 5 - оновлення даних на веб сторінці та повернення в 2 пункт.

3.2 Розробка програмного забезпечення для мікроконтролера

Для розробки програми я вибрав таке середовище як ARDUINO IDE . Це середовище з відкритим програмним кодом дозволяє легко писати код і завантажувати його на плату. В даному програмному середовищі можна використовувати безліч бібліотек.

Також особливістю даного середовища є те , що можна використовувати будь яку плату та датчик ARDUINO також підтримується багато мікроконтролерів та датчиків від сторонніх виробників.

Вихідний код програми мікропроцесорної системи збору даних з різнотипних датчиків на базі ESP8266 з використанням WIFI-модуля ESP8266 я наведу нище.

Але перед цим потрібно встановити всі необхідні бібліотеки а саме , бібліотека для роботи з ESP8266 та Adfruit Unified Sensor. Для цього ви можете відкрити диспетчер бібліотеки в Arduino IDE і виконати пошук у цих двох бібліотеках.

Єдина бібліотека датчиків Adafruit Бібліотека BME280 проекту на базі BME280 необхідний заголовний файл бібліотеки для ESP8266 NodeMCU і ESP32 визначається на першому етапі. ESP8266WebServer.h і бібліотека WebServer.h допомагають нам налаштувати сервер і обробляти вхідні HTTP-запити. Аналогічний дріт . h взаємодіє з будь-яким пристроєм I2C, а не тільки з BME280.

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			35

Бібліотеки Adafruit_BME280.h і Adafruit_Sensor.h - це апаратно-залежні бібліотеки, які обробляють низькорівневі функції, які зображені на рисунку 3.2.

```
#include <Arduino_JSON.h>
#include <ESP8266WebServer.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"
```

Рисунок 3.2 - підключені бібліотеки

Підключення датчиків - в цілому BMP280 , DS18B20 та DHT11 підходять для виміру температури але у цих датчиків точність вимірювання відрізняється. Тому я вирішив для заміру температури використовувати DS18B20, для виміру рівня вологості в повітрі - DHT11. Для визначення атмосферного тиску та висоти над рівнем моря - BMP280, а підключення даних бібліотек зображено на рисунку 3.3.

```
#define DHTTYPE DHT11
// DHT Sensor
uint8_t DHTPin = D6;

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

#define myPeriodic 15
#define ONE_WIRE_BUS D5
#define BMP_SCK D1
#define BMP_MISO D4
#define BMP_MOSI D2
#define BMP_CS D3
```

Рисунок 3.3 - підключення датчиків до мікроконтролера

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		36

Потім ми створюємо об'єкт датчика і змінні для зберігання температури, вологості, тиску та висоти які зображені на рисунку 3.4 відповідно.

```
#define SEALEVELPRESSURE_HPA (1008)

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
Adafruit_BMP280 bme(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
```

Рисунок 3.4 - створення об'єкту датчика та змінних для збереження даних про середовище

Після цього ми створюємо константу для назви та пароля wi-fi мережі що зображено на рис 3.5 відповідно.

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Replace the next line with your API Key
#define ONECALLKEY "REPLACE_WITH_YOUR_API_KEY"
```

Рисунок 3.5- ініціалізація параметрів мережі

Далі проходить ініціалізація API ключа та координат за допомогою якого ми отримуємо дані про погоду зображено на рис 3.6.

```
// Replace the next line with your API Key
#define ONECALLKEY "REPLACE_WITH_YOUR_API_KEY"

// Sample Lat and Lon for Lviv, UA
float myLatitude = 49.8383; //<-----in range to use GPS coordinates
float myLongitude = 24.0232; // Coordinates for Lviv, UA. Change these for your city
```

Рисунок 3.6- ініціалізація ключа

Потім йде ініціалізація сервера , порта та функцій первинного налаштування, зображено на рисунку 3.7.

```

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

  server.handleClient();
  bme.begin(0x76);

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

```

Рисунок 3.7- первинне налаштування

Останні кроки це підключення до локальної мережі, створення сервера , та запуск функції постійного оновлення даних на веб сторінці зображені на рис 3.8 та 3.9.

```

//connect to your local wi-fi network
WiFi.begin(ssid, password);
//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());

```

Рисунок 3.8- підключення до локальної мережі

```

1 server.on("/", handle_OnConnect);
2 server.onNotFound(handle_NotFound);
3 server.begin();
4 Serial.println("HTTP server started");
5
6 }

```

Рисунок 3.9- запуск сервера

Потім проходить постійне виведення та оновлення даних що зображено на рисунку 3.10 та 3.11 відповідно

```
// Serial.print("JSON object = ");
// Serial.println(myObject);
Serial.print("Прогноз: ");
description = JSON.stringify(myObject["daily"][0]["weather"][0]["description"]);
Serial.println(description);
morning = JSON.stringify(myObject["daily"][0]["temp"]["morn"]);
Serial.println(morning);
day = JSON.stringify(myObject["daily"][0]["temp"]["day"]);
Serial.println(day);
evening = JSON.stringify(myObject["daily"][0]["temp"]["eve"]);
Serial.println(evening);
night = JSON.stringify(myObject["daily"][0]["temp"]["night"]);
Serial.println(night);
humidity = JSON.stringify(myObject["daily"][0]["humidity"]);
forecast = "Температура повітря: ранок = " + morning + ", день = " + day + ", вечір = " + evening + ", ніч =
}
else {
  Serial.println("WiFi Disconnected");
}
lastTime = millis();
}
server.handleClient();
```

Рисунок 3.10- виведення даних

```
void handle_OnConnect() {
  float temperature;
  float humidity;
  float pressure;
  float altitude;
  DS18B20.requestTemperatures();
  temperature = DS18B20.getTempCByIndex(0);
  //temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  pressure = bme.readPressure() / 100;
  altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
  Serial.print(" Temperature: ");
  Serial.println(temperature);
  Serial.print(" Humidity: ");
  Serial.println(humidity);
  Serial.print(" Pressure= ");
  Serial.print(pressure);
  Serial.println(" КПа");
  Serial.print(altitude);
  Serial.println(" м");
  server.send(200, "text/html", SendHTML(temperature,humidity,pressure,altitude,forecast));
}
```

Рисунок 3.11- запуск постійного оновлення даних

3.3 Розробка веб сторінки

Веб сторінка розроблена за допомогою мови розмітки гіпертексту HTML та таблиці каскадних стилів CSS. На сторінці були використані шрифти сімейства

									Арк.А
									39
Змн.	Арк.А	№ докум.№	ПідписПі	Дата					

08-23.БДП.033.00.000 ПЗ

Open Sans. Було встановлено кодування UTF-8. В якості картинок було використані зображення в форматі svg. Перевагою такого формату зображень є можливість масштабування без втрати якості.

Веб сторінка виглядає досить мінімалістично інтерфейс не перенавантажений та легкий для сприйняття користувачем. Так як сторінка досить проста то завантаження відбувається досить швидко. Приклад роботи веб сторінки зображений на рисунку 3.12.



Рисунок 3.12 – приклад роботи веб сторінки

Відображення прогнозу погоди відбувається завдяки запиту до сайту <http://api.openweathermap.org>. Приклад запиту зображений на рисунку 3.13.

```
http://api.openweathermap.org/data/2.5/onecall?lat=" +  
(String)myLatitude + "&lon=" + (String)myLongitude +  
&exclude=minutely,hourly&appid=" + ONECALLKEY +  
&units=metric
```

Рисунок 3.13 – приклад запиту

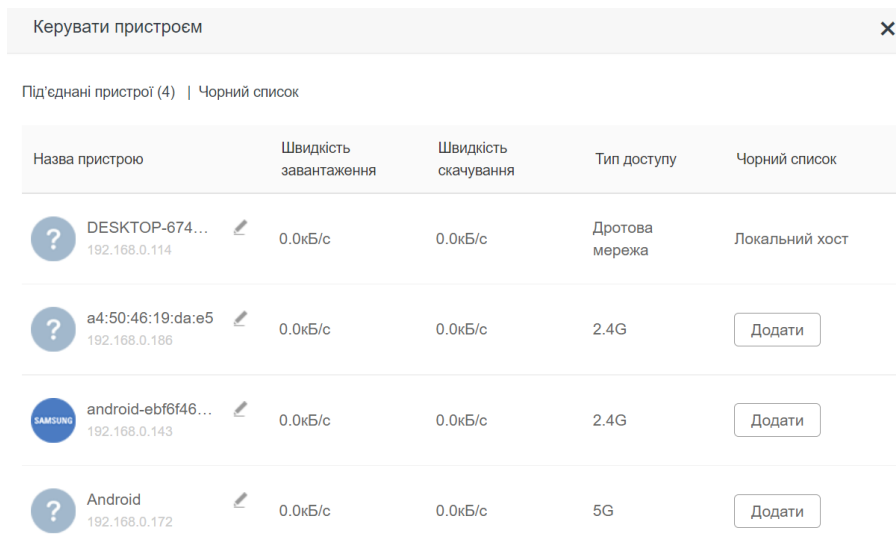
В тіло запиту ми передаємо 3 параметри: myLatitude, myLongitude,

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		40

ONECALLKEY. Де myLatitude - це широта місцезнаходження пристрою, myLongitude – це довгота та ONECALLKEY це наш арі ключ за допомогою якого відбувається зв'язок з сервером, який відповідає за прогноз погоди.

3.4 Підключення приладу до локальної мережі

Після підключення всіх датчиків і завантаження програмного коду в пам'ять мікроконтролера, пристрій повинен автоматично під'єднатися до вказаної в ssid WI-FI мережі. Найчастіше роутер видає динамічну для контролера ip адресу з використанням DHCP сервера. Її можна переглянути в панелі адміністратора вашого роутера як на рисунку 3.14.



The screenshot shows a web interface titled "Керувати пристроєм" (Manage devices) with a close button. Below the title, it says "Під'єднані пристрої (4) | Чорний список" (Connected devices (4) | Blacklist). The main content is a table with the following columns: "Назва пристрою" (Device name), "Швидкість завантаження" (Download speed), "Швидкість скачування" (Upload speed), "Тип доступу" (Access type), and "Чорний список" (Blacklist). There are four rows of devices:

Назва пристрою	Швидкість завантаження	Швидкість скачування	Тип доступу	Чорний список
DESKTOP-674... 192.168.0.114	0.0кБ/с	0.0кБ/с	Дротова мережа	Локальний хост
a4:50:46:19:da:e5 192.168.0.186	0.0кБ/с	0.0кБ/с	2.4G	Додати
android-ebf6f46... 192.168.0.143	0.0кБ/с	0.0кБ/с	2.4G	Додати
Android 192.168.0.172	0.0кБ/с	0.0кБ/с	5G	Додати

Рисунок 3.14 - панель адміністратора роутера

Якщо все виконано вірно то перейшовши по цій адресі з будь якого пристрою що підключений до даної мережі WI-FI ми побачимо таке як на рисунку 3.15 :

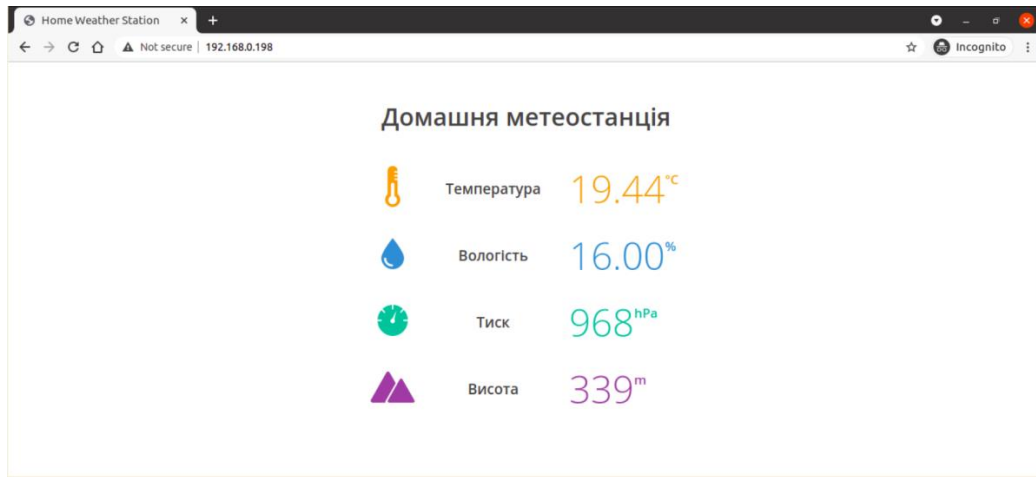


Рисунок 3.15– Соловна сторінка

Також ми можемо зробити наш пристрій доступним в глобальній мережі інтернет. Через те що у нас немає статичної ір адреси, нам потрібно буде скористатися сервісом на кшталт noIP. Ці сервіси потрібні для того щоб закріпити відкритий порт роутера на ір адресі нашого приладу в статичному вигляді. Спочатку потрібно відкрити порт 80. Інтерфейси адміністративної панелі у різних виробників відрізняються, але в моєму випадку так як на рисунку 3.16:

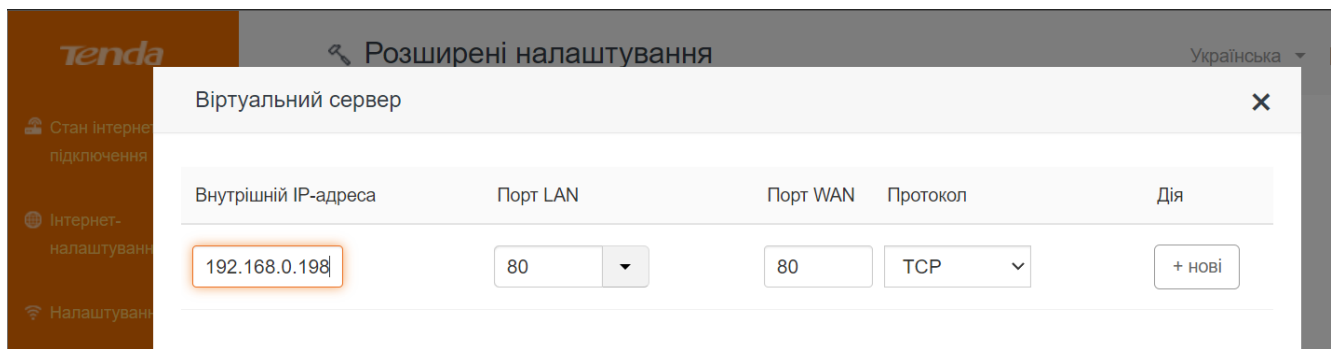


Рисунок 3.16 – Створення віртуального сервера

Далі можна зареєструватися на сервісі NoIP де ми можемо створити тестовий домен для нашого проекту зразок зображений на рисунку 3.17.

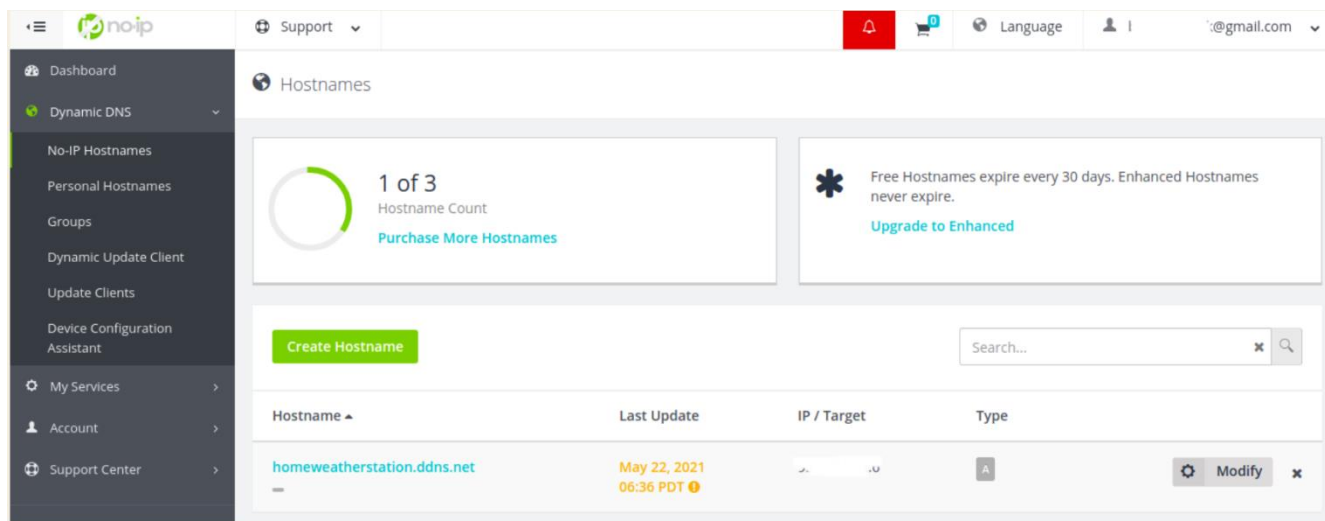


Рисунок 3.17 - головна сторінка NoIP

Завдяки цьому у нас тепер є можливість переходити за цим доменним іменем в мережі інтернет з будь-якого девайсу та в будь-якій точці світу.

Відображення прогнозу погоди, ми можемо реалізувати завдяки такому сервісу як openweathermap.org. Після реєстрації на сайті можна отримати ключ за допомогою якого можна здійснювати запити та отримувати від сервера інформацію про прогноз погоди, а сам ключ зображений на рисунку 3.18.

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.



Key	Name	Status	Actions	Create key
5301343b3157b667d993dab82f71f4e6	Default	Active	 	<input type="text" value="API key name"/>

Рисунок 3.18 - API ключ

API ключ представляє собою унікальний набір символів з випадково згенерованих чисел та літер, який передається при кожному запиті до певної служби пошуку. Потім служба приймає цей ключ, і якщо він дійсний .

Висновки

Результатом моєї дипломної роботи є розробка мікропроцесорної системи збору даних з різнотипних датчиків. Також був виготовлений діючий макет даного пристрою.

У першій частині дипломної роботи було розглянуто мікропроцесорні системи збору даних, типи та різновиди датчиків, інтерфейси. Що в свою чергу надало змогу остаточно вибрати мікроконтролер, на основі якого був розроблений пристрій, та надало більш точну інформацію для подальшої розробки системи в цілому.

У другій частині дипломної роботи, відповідно до технічного завдання була розроблена структурна та функціональна схема мікропроцесорної системи, також були визначені характеристики контролера та датчиків. Визначений склад елементів який був потрібен.

В третій частині дипломної роботи був описаний алгоритм програми в цілому. Також була розроблена прошивка для мікроконтролера. Була розроблена веб сторінка з комфортним та інтуїтивно зрозумілим інтерфейсом. Було проведено первинне налаштування та підключення приладу до локальної мережі.

Розроблений пристрій на основі мікроконтролера та веб сайт відповідає всім вимогам та має певні переваги, такі як: інтуїтивно зрозуміла веб сторінка, низька ціна компонентів для мікропроцесорної системи, простота в використанні та можливість подальшого вдосконалення.

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			44

Перелік джерел посилання

1. Що таке мікропроцесор, мікроконтролер та програмований логічний контролер [Електронний ресурс] – Режим доступу до ресурсу: https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php

2. ЩО ТАКЕ ІНТЕРНЕТ РЕЧЕЙ? [Електронний ресурс] – Режим доступу до ресурсу: <http://iot.lviv.ua/що-таке-інтернет-речей/>

3. Все про ESP 8226 [Електронний ресурс] – Режим доступу до ресурсу:

https://defence-ua.com/video/bezpilotnij_awacs_dlja_zsu_plastun_ta_hortitsja_vid_nvts_infozahist-71.html

4. Уилли Соммер Программирование микроконтроллерных плат Arduino / Freeduino. БХВ - Петербург - 2012.Какие дроны использовать для сельского хозяйства [Електронний ресурс] – Режим доступу до ресурсу:

https://biz.censor.net/resonance/3129115/kakie_drony_ispolzovat_dlya_selskog_o_hozyayistva

5. Що таке мікропроцесор, мікроконтролер та програмований логічний контролер [Електронний ресурс] - Режим доступу до ресурсу: https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.phpКласифікація безпілотних літальних апаратів URL: <http://mino.esrae.ru/pdf/2017/6Sm/1546.doc> (дата звернення: 17.05.2022)

6. Raspberry Pi для начинающих | Занимательная робототехника [Електронний ресурс] - Режим доступу до ресурсу: <http://edurobots.ru/raspberry-pi-dlya-nachinayushhix/> (дата звернення: 27.05.2022)

7. ДСТУ 2226–93 Автоматизовані системи. Терміни та визначення. – К.: УкрНДІССІ, 1994. – 92 с.

						08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата			45

8. Брамм, П. Микропроцессор 80386 и его программирование / П. Брамм, Б. Брамм. - М.: Мир, 1990.

9. Студопедия – макетування в художньому конструюванні [Електронний ресурс] – Режим доступу до ресурсу: <https://studopedia.org/6-7321.html>

					08-23.БДП.033.00.000 ПЗ	Арк.А
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		46

Додаток А

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ О. Д. Азаров

“ ___ ” _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврського дипломного проекту
«Мікропроцесорна система збору даних з різнотипних датчиків»
08-23.БДП.037.00.000 ПЗ

Науковий керівник к.т.н., доц. каф. ОТ

_____ Богомолов С.В.

Студент групи 1КІ-20МС

_____ Бугай Д.О.

Вінниця 2022

1 Найменування та область застосування

Робоча назва проекту «мікропроцесорна система збору даних з різнотипних датчиків», розробляється для навчання студентів створювати та проектувати комп'ютерні системи.

2 Основи для розробки

Основою для розробки є дисципліни: комп'ютерна логіка, комп'ютерна електроніка, комп'ютерні системи.

3 Мета та призначення розробки

Експлуатаційне призначення розробки - створення мікропроцесорної системи збору даних з різнотипних датчиків.

4 Етапи БДП та очікувані результати

Робота виконується в п'ять етапів, що наведені в таблиці 4.1.

Таблиця А.1 - Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	14.02.22	28.02.22	Вступ
2	Огляд і аналіз мікропроцесорних систем збору даних з різнотипних датчиків	15.02.22	14.03.22	розділ 1
3	Проектування системи	15.03.22	28.03.22	Розділ 2
4	Реалізація програмного та апаратного забезпечення	29.03.22	11.04.22	Розділ 3
5	Оформлення пояснювальної записки	12.04.22	25.04.22	ПЗ, презентація
6	Оформлення пояснювальної записки та ілюстративного матеріалу	03.05.21	29.05.21	
7	Перевірка якості виконання бакалаврського проекту та усунення недоліків	30.05.22	30.05.22	

5 Матеріали, що подаються до захисту бакалаврської дипломної роботи

Пояснювальна записка містить графічні і ілюстративні матеріали, відзив наукового керівника, рецензія опонента, анотації українською та іноземною мовами, довідка про відповідність оформлення бакалаврської дипломної роботи діючим вимогам.

6 Порядок контролю виконання та захисту БДП

Виконання етапів графічної та розрахункової документації БДП контролюється науковим керівником згідно зі встановленими термінами. Захист БДП відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

7 Вимоги до оформлення БДП

Вимоги викладені:

- ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

- ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

- ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

- документами на які посилаються у вище вказаних.

Технічне завдання до виконання отримав _____ Бугай Д.О.

Додаток Б

Скетч для програмування мікроконтролера та веб-сторінки

```
#include <Arduino_JSON.h>
#include <ESP8266WebServer.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"

#define DHTTYPE DHT11
// DHT Sensor
uint8_t DHTPin = D6;

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

#define myPeriodic 15
#define ONE_WIRE_BUS D5
#define BMP_SCK D1
#define BMP_MISO D4
#define BMP_MOSI D2
```

```

#define BMP_CS D3

#define SEALEVELPRESSURE_HPA (1008)

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
Adafruit_BMP280 bme(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Replace the next line with your API Key
#define ONECALLKEY "5301343b3157b667d993dab82f71f4e6"

// Sample Lat and Lon for Lviv, UA
float myLatitude = 49.8383; //<-----in range to use GPS
coordinates

float myLongitude = 24.0232; // Coordinates for Lviv, UA. Change these for
your city

ESP8266WebServer server(80);
unsigned long lastTime = 0;
unsigned long timerDelay = 10000;
String jsonBuffer;
String forecast, morning, day, evening, night, humidity, description;

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

```

```
delay(100);

server.handleClient();
bme.begin(0x76);

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

//connect to your local wi-fi network
WiFi.begin(ssid, password);
//check wi-fi is connected to wi-fi network
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() {
  if ((millis() - lastTime) > timerDelay) {
```

```

// Check WiFi connection status
if(WiFi.status()== WL_CONNECTED){
    String serverPath = "http://api.openweathermap.org/data/2.5/onecall?lat=" +
(String)myLatitude + "&lon=" + (String)myLongitude +
"&exclude=minutely,hourly&appid=" + ONECALLKEY + "&units=metric";

    jsonBuffer = httpGETRequest(serverPath.c_str());
    //Serial.println(jsonBuffer);
    JSONVar myObject = JSON.parse(jsonBuffer);

    // JSON.typeof(jsonVar) can be used to get the type of the var
    if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;
    }

    // Serial.print("JSON object = ");
    // Serial.println(myObject);
    Serial.print("Прогноз: ");
    description =
JSON.stringify(myObject["daily"][0]["weather"][0]["description"]);
    Serial.println(description);
    morning = JSON.stringify(myObject["daily"][0]["temp"]["morn"]);
    Serial.println(morning);
    day = JSON.stringify(myObject["daily"][0]["temp"]["day"]);
    Serial.println(day);
    evening = JSON.stringify(myObject["daily"][0]["temp"]["eve"]);
    Serial.println(evening);

```

```

    night = JSON.stringify(myObject["daily"][0]["temp"]["night"]);
    Serial.println(night);
    humidity = JSON.stringify(myObject["daily"][0]["humidity"]);
    forecast = "Температура повітря: ранок = " + morning + ", день = " + day
+ ", вечір = " + evening + ", ніч = " + night + ", вологість = " + humidity + ", " +
description + ".";
    }
    else {
        Serial.println("WiFi Disconnected");
    }
    lastTime = millis();
}
server.handleClient();
}

```

```

void handle_OnConnect() {
    float temperature;
    float humidity;
    float pressure;
    float altitude;
    DS18B20.requestTemperatures();
    temperature = DS18B20.getTempCByIndex(0);
    //temperature = dht.readTemperature();
    humidity = dht.readHumidity();
    pressure = bme.readPressure() / 100;
    altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
    Serial.print(" Temperature: ");
    Serial.println(temperature);
}

```

```

Serial.print(" Humidity: ");
Serial.println(humidity);
Serial.print(" Pressure= ");
Serial.print(pressure);
Serial.println(" KPa");
Serial.print(altitude);
Serial.println(" m");
server.send(200, "text/html",
SendHTML(temperature,humidity,pressure,altitude,forecast));
}

String httpGETRequest(const char* serverName) {
    HTTPClient http;

    // Your IP address with path or Domain name with URL path
    http.begin('192.168.0.106');

    // Send HTTP POST request
    int httpResponseCode = http.GET();

    String payload = "{}";

    if (httpResponseCode>0) {
        // Serial.print("HTTP Response code: ");
        // Serial.println(httpResponseCode);
        payload = http.getString();
    }
    else {

```

```

    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
// Free resources
http.end();

return payload;
}

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

```

Скетч побудова веб-сторінки

```

String SendHTML(float temperature, float humidity, float pressure, float altitude,
String forecast){
    String ptr = "<!DOCTYPE html>";
    ptr += "<html>"; ptr += "<head>";
    ptr += "<title>Home Weather Station</title>";
    ptr += "<meta name='viewport' content='width=device-width, initial-scale=1.0'
charset='UTF-8'></meta>";
    ptr += "<link
href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600'
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 += "</head>";
```

```
ptr += "<body>";
```

```
ptr += "<h1>Домашня метеостанція</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-
2h4.084V6.063C16.06,2.715,13.345,0,9.996,0H9.313C5.965,0,3.252,2.715,3.25
2,6.063v30.982";
```

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

```
ptr += "</div>";
```

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

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

```
ptr +=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='M14.618,0C14.618,0,0,17.95,0,26.022C0,34.096,6.544,40.64,14.618,40.64s1
4.617-6.544,14.617-14.617'";
```

```
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'>Вологість</div>";
```

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

```
ptr +=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-
0.202C14.39,0.279,9.936,2.256,6.544,5.385l3.576,3.577c0.391,0.391,0.391,1.02
4,0,1.414";

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'>Тиск</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'>Висота</div>";
```

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

```
ptr += (int)altitude;
```

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

```
ptr += "</div>";
```

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

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

```

String SendHTML(float temperature, float humidity, float pressure,float
altitude, String forecast){

String ptr = "<!DOCTYPE html>";

ptr +="<html>";

ptr +="<head>";

ptr +="<title>Home Weather Station</title>";

ptr +="<meta name='viewport' content='width=device-width, initial-scale=1.0'
charset='UTF-8'></meta>";

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

ptr += "<body>";

ptr += "<h1>Домашня метеостанція</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'>Температура</div>";

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

ptr += temperature;

ptr += "°C</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='M14.618,0C14.618,0,0,17.95,0,26.022C0,34.096,6.544,40.64,14.618,40.64s1
4.617-6.544,14.617-14.617'";
```

```
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'>Вологість</div>";
```

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

```
ptr +=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-
0.202C14.39,0.279,9.936,2.256,6.544,5.385l3.576,3.577c0.391,0.391,0.391,1.02
4,0,1.414";
```

```
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'>Тиск</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'>Висота</div>";
```

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

```
ptr += (int)altitude;
```

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

```
ptr += "</div>";
```

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

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

```
ptr += "<img src='https://www.svgrepo.com/show/212029/rain-weather.svg'
width='55' height='55'>";
```

```
ptr += "</div>";
```

```
ptr += "<div class='side-by-side text'>Прогноз погоди</div>";

ptr += "<div>";

ptr += forecast;

ptr += "</div>";

ptr += "</div>";

ptr += "</div>";

ptr += "</body>";

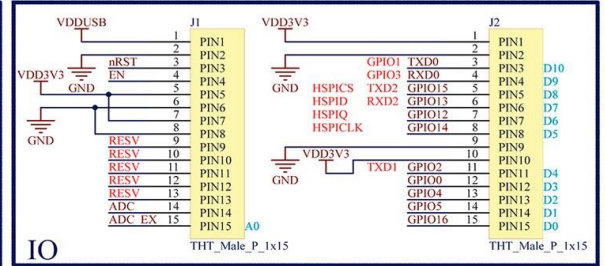
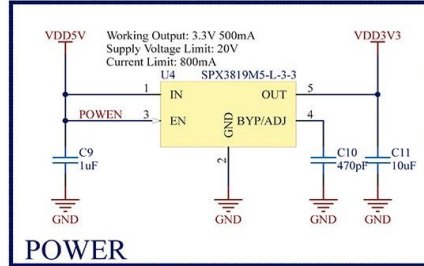
ptr += "</html>";

return ptr;

}
```

Додаток В

Структура схема плати ESP8266 NodeMCU



NODE MCU

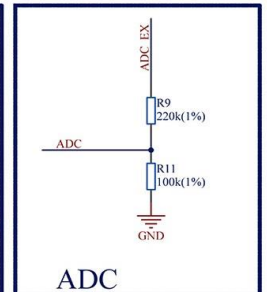
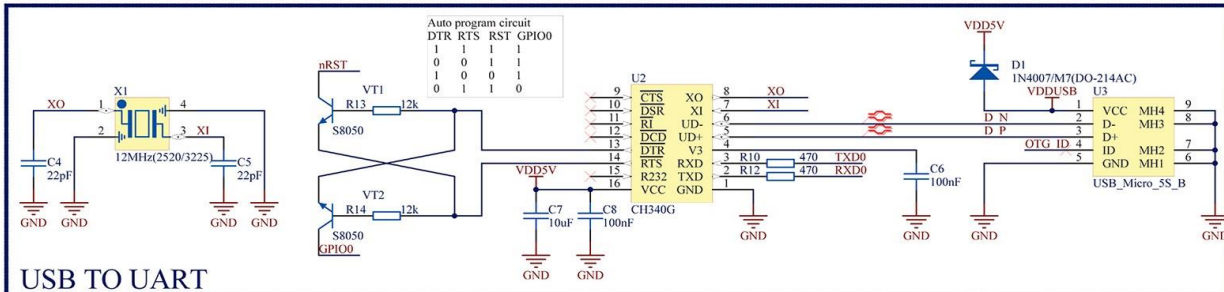
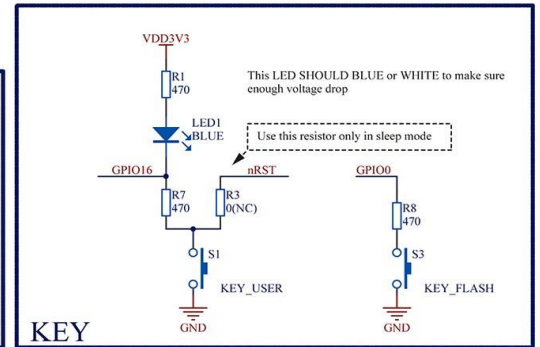
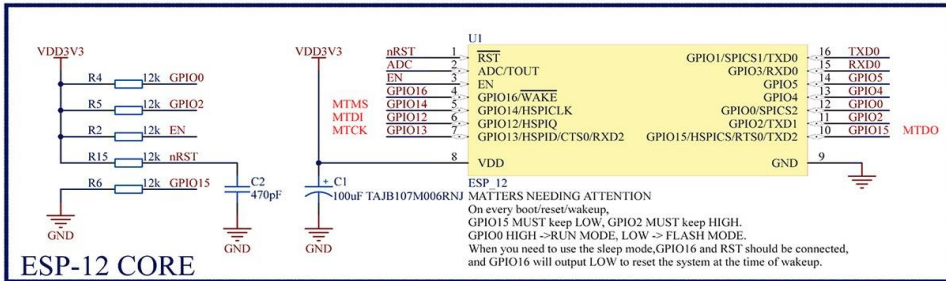


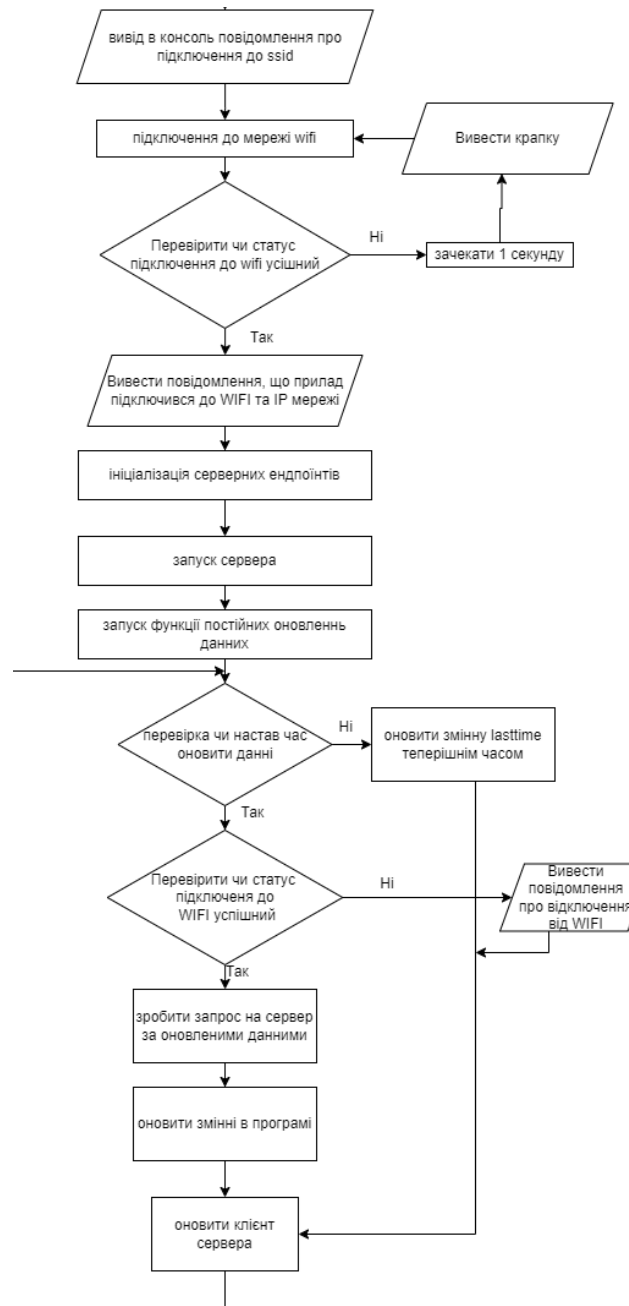
Рисунок В - структура схема плати ESP8266 NodeMCU

Додаток Г

Зовнішній вигляд повного алгоритму програми



Рисунок Г - Зовнішній вигляд повного алгоритму програми



Продовження рисунка Г

Додаток Д
 ПРОТОКОЛ
 ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
 НА НАЯВНІСТЬ ТЕКСТОВИХ
 ЗАПОЗИЧЕНЬ

Назва роботи: Мікропроцесорна система збору даних з різнотипних датчиків

Тип роботи: бакалаврська дипломна робота

(БДР, МКР)

Підрозділ кафедра обчислювальної техніки

(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 91,8% Схожість 9,2%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.

(підпис)

(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Бугай Д. О.

(підпис)

(прізвище, ініціали)

Керівник роботи _____ Богомолів С. В.

(підпис)

(прізвище, ініціали)