

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____Олексій СМІРНОВ
“_____” _____2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення інтелектуальної системи «розумний
годинник» з автентифікацією даних через мобільний додаток”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____Кічура М.Г.
«_____» _____2026р.

Керівник проекту
кандидат технічних наук, доцент
_____Минайленко Р.М.
«_____» _____2026р.
Рецензент _____

АНОТАЦІЯ

Кічура М.Г. Програмне забезпечення інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток. 122 Комп'ютерні науки. Центральнотураїнський національний технічний університет. Кропивницький. 2026.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток.

Метою розробки є Програмне забезпечення інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток.

Результат роботи – програмна реалізація інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Arduino IDE, Kotlin.

Ключові слова: комп'ютерні науки, Arduino, Kotlin.

ABSTRACT

Kichura M.G. Software for an intelligent "smart watch" with data authentication via a mobile application. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026

In this bachelor's qualification the software which is intended for intelligent "smart watch" with data authentication via a mobile application is developed.

The purpose of the development is the software for an intelligent "smart watch" with data authentication via a mobile application.

The result of this work is the software implementation of an intelligent smart watch system with data authentication through a mobile application.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on a PC with Windows 10/11.

The program is developed in the environment Arduino IDE, Kotlin.

Keywords: computer science, Arduino, Kotlin

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
“ _____ ” _____ 2026 року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА
ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

Кічури Максима Григоровича
(прізвище, ім’я, по батькові)

1. Тема роботи Програмне забезпечення інтелектуальної системи
«розумний годинник» з автентифікацією даних через мобільний додаток

2. Керівник роботи Минайленко Роман Миколайович канд. техн. наук, доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти №116-02 від 06.02.2026 року

3. Строк подання роботи до захисту 22.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи Метою роботи є розробка
програмного забезпечення інтелектуальної системи «розумний годинник» з
автентифікацією даних через мобільний додаток

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання 06.02.2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.05.2026 р.	
8.	Попередній захист роботи	22.05.2026 р.	

Дата видачі завдання
06.02.2026 р.

Підпис керівника

Минайленко Р.М.
(прізвище та ініціали)

Завдання прийнято до виконання
06.02.2026 р.

Підпис здобувача

Кічура М.Г.
(прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи	6
1.2 Область застосування	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування....	21
2.3 Розгорнута постановка завдання	29
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	31
3.1 Опис функціонування системи	31
3.2 Розробка структурної схеми	34
3.3 Розробка функціональної схеми	38
3.4 Розробка діаграми процесів	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	49
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	50
6 ОСНОВНІ ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

					ВКРБ-122.26.0016.00.00.ПЗ										
Вим.	Арк.	№ докум.	Підпис	Дата	Програмне забезпечення інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток				Літ		Аркуш	Аркушів			
Розробив		Кічура М.Г.										1	62		
Перевір		Минайленко Р.М.													
Н. Контр.		Коваленко А.С.													
Затверд.		Сидіненко О.А.			ЦНТУ КН-22										

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

Opensource – Вільнодоступне ПЗ

IoT – Інтернет речей (Internet of Things)

ESP32 – мікроконтролер серії Espressif Systems 32-bit

API – Application Programming Interface

ПЗ – Програмне забезпечення

HTTP – HyperText Transfer Protocol

Smart Watch – Розумний годинник

і т.д. – і так далі

К6ПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						2
Вим	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Розвиток мікроконтролерів та недорогих IoT-платформ зробив можливим створення компактних електронних пристроїв для побутового, навчального й демонстраційного використання. У межах цієї роботи такі підходи застосовано для побудови системи «розумний годинник», яка поєднує мікроконтролерну частину, локальний сервер і мобільний застосунок для автентифікації користувача.

Історично розвиток годинникових пристроїв пройшов шлях від сонячних і механічних конструкцій до кварцових та електронних систем. Для цієї роботи важливим є не сам історичний розвиток годинників, а перехід доелектронних пристроїв, які можуть взаємодіяти з програмним забезпеченням, датчиками та мобільними застосунками.

У середньовічній Європі з'явилися перші механічні годинники, які працювали за допомогою механізмів із колесами та гирями. Вони ставилися на дзвіниці церков, міських ратуш та на важливих громадських будівлях. Адже були великими, громіздкими та потребували постійного заправлення, однак вони стали важливими для соціального життя того часу, бо допомагали синхронізувати час для всіх громадян.

Прорив у годинниковій техніці відбувся в XVI столітті, коли були винайдені пружинні годинники. Вони дозволяли значно зменшити розміри, зробивши їх більш мобільними. Перші такі годинники були настільними, але з часом з'явилися портативні версії, які могли носити з собою заможні люди. Підвищення точності та зручності механізмів дозволило людям ще більше покладатися на цей інструмент у повсякденному житті.

Наприкінці XX століття на зміну механічним годинникам прийшли кварцові годинники, які використовували електричні імпульси для регулювання часу. Кварцові годинники відрізняються високою точністю, низьким рівнем обслуговування та довгим терміном роботи від батареї. Ці годинники стали дуже популярними завдяки їхній доступності та надійності, і до сьогодні вони займають провідне місце на ринку.

Традиційні годинники еволюціонували від простих механічних пристроїв відліку часу до складних електронних систем, здатних виконувати багато функцій,

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						3
Вим	Арк.	№ докум.	Підпис	Дата		

окрім простого відображення часу.

Завдяки прогресу в технології мікроконтролерів та інтеграції датчиків зараз годинник має функції на рівні смартфона.

Розумний годинник - це розумний наручний пристрій, який окрім вимірювання часу, виконує додаткові функції: відтворення музики, прийом сповіщень та дзвінків, відстеження маршрутів з допомогою GPS, збір інформації з вбудованих та зовнішніх датчиків. Існують два види розумних годинників, за цільовим призначенням та за автономністю.

Різниця між звичайним і розумним годинником дуже суттєва, в той час як звичайний годинник просто показує час, розумний годинник має більше можливостей таких як дзвінки або відправлення повідомлень через СМС та месенджери.

Окрему групу аналогів становлять open-source проєкти, опубліковані на технічних форумах, GitHub та освітніх платформах. Зазвичай такі рішення створюються ентузіастами, студентами або інженерами з метою демонстрації роботи мікроконтролерів, датчиків, дисплеїв та базової взаємодії з програмним забезпеченням. Їхньою перевагою є відкритість коду й низька собівартість, однак для повторення таких рішень користувач повинен мати базові навички роботи з Arduino IDE, ESP32 та електронними компонентами.

В більшості проєктів є гайди або відеоуроки як їх правильно встановити та налаштувати, але це все буде важко зрозуміти звичайній людині, тому що в більшості відео використовується в основному професійна термінологія та вони направлені в більшості на спеціалістів або таких же ентузіастів, які хочуть вдосконалити або на цій базі зробити свій «розумний годинник», але більшість з таких проєктів не мають будь якої взаємодії з мобільним телефоном.

Актуальність теми полягає в необхідності створення простого навчального прототипу розумного годинника, який демонструє взаємодію між пристроєм, локальним HTTP-сервером і мобільним застосунком. Такий підхід дозволяє перевірити логіку автентифікації користувача, структуру API, роботу інтерфейсу та можливість подальшого перенесення системи на ESP32-сумісну апаратну платформу.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						4
Вим	Арк.	№ докум.	Підпис	Дата		

Мета й завдання дослідження. Метою роботи є програмне забезпечення інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток. Головною метою є створення простого та зручного пристрою, та локальний сервер з демонстрацію роботи годинника.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем «розумний годинник» з автентифікацією даних через мобільний додаток.
- Дослідження основних принципів роботи розумних годинників.
- Створення ПЗ для локального сервера-демонстрації розумного годинника з автентифікацією через мобільний додаток.
- Створення ПЗ для годинника та мобільного додатку.
- Перевірка функціональності локального сервера та розумного годинника з мобільним додатком.

Практична цінність отриманих результатів. Результати цієї роботи мають велике практичне значення в навчальній галузі. Розроблене ПЗ може бути використане в як навчальний стенд-демонстрація принципів IoT-систем, також як взаємодія Android-застосунку з локальним сервером, firmware-підхід для ESP32 та моделювання показників сенсорів. Також система може виступити практичним прикладом програмної взаємодії мобільного додатку з пристроєм як у вигляді локального HTTP-сервера так і в практичному в вигляді самого пристрою для виконання лабораторної роботи студентом.

Таким чином, у результаті виконання роботи буде створено функціональний розумний годинник, який стане не тільки практичним інструментом для вимірювання часу, що забезпечить зручність використання та контроль за часом через мобільний телефон.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ СИСТЕМИ

1.1 Призначення Системи

Розроблювана в бакалаврській роботі система розумний годинник яку можна охарактеризувати як електронний розумний пристрій - який впливає на різні аспекти нашого повсякденного життя. Серед численних пристроїв, електронні розумні прилади набули широкої популярності завдяки своєму функціоналу та простоті в використанні.

На мою думку завдяки розвитку технології мікроконтролерів, та платформам, таких як Arduino, ми маємо багату варіативність розумних пристроїв в тому числі й годинників, які виходять набагато за рамки відстеження часу. Ці пристрої містять різні функції, які підвищують зручність для користувачів, за роки скільки вже існують подібні пристрої, ми маємо розумні годинники які вже можуть конкурувати з телефонами що робить їх важливою частиною сучасних цифрових приладів для нашого сьогодення.

Розроблена програма призначена для реалізації системи “розумного годинника”, через демонстрацію шляхом локального HTTP API сервер на мові C++ для того щоб мати можливість продемонструвати розроблене ПЗ без наявного реального пристрою в режимі реального часу. Система розумний годинник в демонстрації може реалізовувати ті ж самі функції так само як і реальний пристрій. А саме: відображення стану пристрою, автентифікація, отримання даних, синхронізація часу, встановлення будильника та блокування пристрою.

Основною функцією годинника є відображення часу, яке залишається його основним призначенням. Сучасні реалізації цих пристроїв покладаються на модуль годинника реального часу (RTC) для забезпечення точного відліку часу. На відміну від традиційних механічних годинників, електронні версії пропонують більшу гнучкість у синхронізації та налаштуванні часу. Інтеграція модулів RTC із системами на основі Arduino забезпечує автоматичну корекцію та оновлення часу через пристрій чи інтернет та підвищує надійність, що робить ці пристрої незамінними для сучасної людини.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		6

Ще одна важлива функція сучасних розумних годинників – це можливість вимірювати температуру повітря в приміщенні. Ця функція зазвичай досягається завдяки використанню різних датчиків температури, які надають дані про температуру з високою точністю. Зібрані дані можуть відображатися безпосередньо на інтерфейсі годинника, пропонуючи користувачам цінну інформацію без додаткових пристроїв.

Система складається з трьох частин. Перша частина: локальний симулятор на ПК написаний мовою C++ який має всі функції “системи розумний годинник” з яким можна взаємодіяти за допомогою додатку. Сам симулятор запускається на ПК і з ним можна повністю взаємодіяти через мобільний додаток. Друга частина: мобільний застосунок написаний в програмі Kotlin, він відіграє роль клієнта для автентифікації та керування пристроєм. Третя частина: Arduino скетч розроблений під мікроконтролер ESP-32.

Демонстрація має всі функції які притаманні розумному годиннику, зміна часу, температури, зміна кількості кроків, рівень зарядки, будильник та статус автентифікації. Інтерфейс мінімалістичний, і вся інформація з функціями годинника відображаються на OLED екрані.

Функціонально система розділяється на 3 функції, інформаційну, керувальну та демонстраційну. Інформаційна функція складається в отриманні та відображенні інформації системи: часу, дати, дня тижня, кількості кроків, рівня заряду батареї, температури, будильника та стану автентифікації. Керувальна функція забезпечує функціонально частину програми, і вона відповідає за синхронізацію часу, встановлення будильника, блокування та автентифікацію годинника. Демонстраційна функція може продемонструвати можливості роботи програми без фізичного пристрою на комп’ютері.

Особливості розроблюваної системи наступні. Поєднання фізичної логіки розумного годинника з локальною демонстрацією яку можна запустити та взаємодіяти з нею на основі ПК. Система працює та демонструє роботу та повністю взаємодіє з мобільним додатком. Мобільний додаток виступає як автентифікатор за задачею до роботи, основною функцією якого являється автентифікація з системою розумного годинника.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						7
Вим	Арк.	№ докум.	Підпис	Дата		

Застосунок працює як клієнт до локального сервера. Автентифікація системи виконується через мобільний додаток за допомогою PIN коду, після чого застосунок надсилає запит до годинника.

1.2 Область застосування

Розроблене програмне забезпечення доцільно розглядати насамперед як навчальний і демонстраційний прототип. Його можна використовувати для пояснення принципів взаємодії мобільного застосунку з локальним API, моделювання роботи IoT-пристрою, базової автентифікації користувача та програмування ESP32-сумісних мікроконтролерів.

Систему розумний годинник за моєю думкою можна адаптувати під різні лабораторні роботи будь яким чином в залежності від навчальної дисципліни. Наприклад, на її основі можна демонструвати роботу локальних серверів, розробку мобільних додатків для пристроїв, обробку помилок, роботу з Android застосунками, базові навички з Інтернету речей, програмування мікроконтролерів та розробку пристроїв за допомогою Arduino. Слід зазначити що це відкриває багату варіативність та можливості багатьох варіантів для використання застосунку в навчальних цілях.

Подальший розвиток системи розумний годинник з автентифікацією через мобільний застосунок, можна покращити наступними функціями їх можна розподілити на декілька груп: сповіщення, зв'язок та керування смартфоном, функції для моніторингу здоров'я, активність та спорт, музика, медіа та розваги, платежі та безпека.

Почнемо з того що всі покращення та нові функції будуть корисні для навчальних цілей зазначених вище, бо вони дають більший спектр для дослідження студентом різних навчальних дисциплін, або ж більше охоплення інших навчальних предметів де б могла використовуватись одна з нижческазаних функцій як предмет для навчання.

Сповіщення, зв'язок та керування смартфоном: данна група функцій може покращити пристрій додаючи такі функції як синхронізація з смартфоном та вивід на дисплей годинника всі вхідні сповіщення, такі як повідомлення, події з календаря, нагадування. Також можна додати можливість виконувати дії як: відповідь на

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						8
Вим	Арк.	№ докум.	Підпис	Дата		

повідомлення, скидати або приймати дзвінки. Це робить розумний годинник більш незалежним пристроєм та також більш зручним в різних ситуаціях які можуть бути в повсякденному житті.

Функції для моніторингу здоров'я: тут назва групи говорить сама за себе та й це також ключова функція яка стає причиною купівлі розумного годинника. Функції здоров'я в смарт-годинниках особливо цінні для людей похилого віку, хворих на хронічні захворювання серцево-судинної або дихальної системи, а також тих, хто хоче тримати під контролем своє здоров'я, для цих людей дуже важливо знати наприклад пульс та ритм серця, або ж простий моніторинг та цілодобовий аналіз стану здоров'я. В функції моніторингу можна зазначити наступне: вимірювання пульсу та ритму серця, відстежування рівню стресу, варіабельність пульсу та дихання, жіночий календар та нагадування про приймання ліків.

Активність та спорт: один з основних напрямків роботи розумного годинника, в девайс можна вбудувати декілька спортивних режимів: біг, велоспорт, плавання, йога, лижі, альпінізм, триатлон та інші. Також годинник може показувати пройдену відстань, темп ходьби, серцевий ритм під час тренувань та інші метрики в режимі реального часу або як статистику. Ще можна зробити, доступний аналіз тренувань, поради з навантажень, алгоритми поліпшення результатів та інше – це робить годинник найкращим інструментом для любителів спорту, професіоналів і просто активних людей.

Музика, медіа та розваги: з основного можна використовувати для керування музикою на телефоні: перемикає треки, регулювати гучність, запускати пісні. Також можна імпортувати класичні 2д ігри в розумний годинник, такі як змійка та пінг-понг.

Платежі та безпека: це дає можливість розумному годиннику виконувати безконтактну оплату в магазинах (із наявністю NFC-модуля). У просунутих моделях смарт-годинників також є функція збереження електронних квитків, посадкових талонів та проїзних. Багато пристроїв підтримують системи SOS – при падінні або збоях серцевого ритму годинник автоматично відправляє сигнал тривоги та геолокацію близьким це дуже корисна функція безпеки для будь кого яка здатна врятувати життя в надзвичайних ситуаціях.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						9
Вим	Арк.	№ докум.	Підпис	Дата		

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень

На ринку та серед пересічних людей, доступний широкий вибір Смарт годинників, розумних годинників, та будь яких інших схожих пристроїв та систем з мобільним додадком або без нього. В користувачів є багато вибору від моделей які можна придбати в магазині техніки до різних саморобних пристроїв, з різними функціями характеристиками та можливостями які відповідають вимогам користувача які залежать від їх вподобань та потреб.

Одним з ключових переваг розумних електронних годинників є їх зручність і практичність. Інтегруючи кілька функцій в одному пристрої, користувачі можуть заощадити час і зменшити залежність від окремих гаджетів. Можливість контролювати час, відстежувати температуру, бездротово заряджати пристрої та встановлювати будильники через з'єднання Bluetooth спрощує повсякденні справи та підвищує загальну продуктивність. Крім того, автоматизація певних завдань, таких як розклад будильників, моніторинг температури, сприяє більш ефективному та організованому способу життя.

Почнемо з не дуже відомого бренда Amazfit це сімейство годинників з'явилося на нашому ринку відносно нещодавно, але вони вже почали покорювати серця людей своєю простотою та ціною яка по зрівнянням з їх конкурентами найдешевша серед всіх.

Годинник Amazfit представляє з себе недорогий пристрій для повсякденного використання. Головна відмінність його від інших аналогів за такою ж ціною – наявність GPS модуля для повноцінної навігації без смартфона, також є наявність окремого спортивного режиму для плавання.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						10
Вим	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.1 – Розумний годинник Amazfit

При поєднанні зі смартфоном через Bluetooth на дисплей виводяться повідомлення про виклики та сповіщення з різних програм. Цей розумний годинник не перевантажений зайвими функціями, та має все, що потрібно для базового використання.

З мінусів:

- GPS не завжди працює ідеально.
- Екран не надто яскравий, що може ускладнити використання в сонячний день.
- Мінімалістичний дизайн який не всім до вподоби.

Переваги Amazfit:

- Приємна та дешева ціна для користувача.
- Робота батареї та автономність до 14 днів.
- Підтримка Android та iOS.

Взагалом моделі бренду Amazfit доволі дешеві, та мають дуже багато функцій які будуть цікавити багато користувачів яких зацікавить данна модель в основному через свою ціну, вони можуть конкурувати на рівні з дуже відомими брендами по типу Apple

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						11
Вим	Арк.	№ докум.	Підпис	Дата		

або Samsung, але через те що Amazfit родом з Китаю, їм дуже важко залишатися конкурентноспроможними з іншими китайськими брендами по типу Xiaomi які зарекомендували себе як бренд в якого надійні та дешеві пристрої.

Наступним в огляді у нас йде годинник від Apple, сучасний годинник з серії розумних годинників від Apple. Який вирізняється від інших якістю та наявністю багатьох функцій. Від минулої моделі він відрізняється багатьма покращеннями такими як OLED-екран з широкими кутами огляду, більш яскравим екраном, та трохи більшим за минулу модель екраном. Цей годинник має товщину лише 9,7 мм, що робить його дуже тонким в порівнянні з минулим пристроєм. Збільшений дисплей – найбільший серед усіх годинників Apple що дозволяє відображати додаткову лінію тексту на екрані, що полегшує користувачам миттєву перевірку електронних листів і повідомлень з повним вмістом. Також дисплей, та корпус мають більш округлені кути, ніж раніше.



Рисунок 2.2 – Розумний годинник Apple

Годинник чудово інтегрується в екосистему з іншими гаджетами від Apple, що дозволяє дуже просто користуватися годинником коли в тебе більшість пристроїв від цього ж бренду. В пристрої вбудовані датчики для вимірювання ЕКГ, рівня кисню в крові та моніторинг сну. Всі ці функції допомагають користувачу не тільки бути на

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		12

зв'язку, також він пропонує слідкувати за своїм здоров'ям.

З додатками, які підтримують Apple, можна без проблем контролювати всі параметри, від будильника до емоційного стану. А ще наявний голосовий помічник Siri який може дати відповідь на важливе питання або нагадати щось користувачу, наприклад те що ви забули піти в спортзал.

Тепер перейдемо до характеристик годинника Apple.

Зручний інтерфейс та взаємодія з іншими пристроями Apple.

Набагато легчий за минулу модель з можливістю придбання пристрою в титановому корпусі який буде ще на 10% легчий чим базова версія.

Найкращий рівень захисту IP6X який дозволяє не знімати його в душі чи під час плавання, також пристрій водостійкий на глибині до 50 метрів.

Пристрій містить більшу зарядну котушку, що забезпечує швидшу зарядку акумулятора до 80% всього за 30 хвилин.

Процесор Apple S10 з 4-ядерним нейронним блоком для функцій штучного інтелекту.

Новітній застосунок Vitals для перевірки нічних показників, тренувальних навантажень і даних про вагітність.

Завдяки новій версії системи watchOS пристрій має можливість призупиняти й додатково налаштовувати цілі й активності користувачем.

Також з появою нової операційної системи вона принесла нові функції до годинника, такі як віджети які тепер відображатимуть активність у реальному часі. Також став доступним набір функцій для відстеження стану здоров'я, в додаток до наявних вже існуючих корисних функцій з минулих моделей, такими як eSIM або Автоматичне відстеження того як ви впали з висоти чи потрапили в надзвичайну ситуацію, тоді автоматично пристрій повідомляє ваше місцезнаходження для екстрених служб.

На мою думку пристрої Apple можна розглядати як один із найбільш функціонально розвинених прикладів комерційних смарт-годинників.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						13
Вим	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.3 – Розумний годинник Samsung

Переходимо до наступного годинника який не поступається функціями та інноваціями з Apple але на відміну від них ціна в цього бренду не дуже сильно кусається і мова йде про Samsung, це новинка серед розумних годинників Samsung яка має купу нововведень

Розумний годинник Samsung отримав нову ОС, покращений моніторинг здоров'я завдяки штучному інтелекту від Samsung. Пристрій вперше дозволяє відстежувати кінцеві продукти глікації (AGEs), аналіз складу тіла (BIA) для кращого розуміння здоров'я користувача, а двочастотна система точніше відстежує місцеперебування. В додаток до функцій минулих лінійок прослідковується те що можливо стане нормою в майбутніх моделях Samsung, а саме Galaxy AI, який з кожним

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		14

новим пристроєм набирає все більше і більше функцій. У годиннику використовується новий процесор Exynos W1000, який надає 2 ГБ оперативної пам'яті та 32 ГБ вбудованої пам'яті. Що є дуже вагомим плюсом в порівнянні з іншими годинниками конкурентами.

За вищеперечислинною інформацією це вже не просто годинник а повноцінний смартфон на зап'ясті, також Samsung як і Apple отримав такі нововведення, як новий алгоритм штучного інтелекту для відстеження сну, функцію виявлення апное уві сні, а також моніторинг серцевого ритму та повідомлення про нерегулярний пульс.

Також Samsung зазначає, що в пристрої дебютує новий сенсор BioActive, який є більш точним, ніж раніше. Пристрій має титановий корпус, водонепроникність 10 ATM, налаштовану кнопку Quick Button і нову динамічну систему вушок для додаткового комфорту. З іншого боку, Samsung має більш пом'якшений дизайн та менші розміри. Пристрій не дуже схожий на минулорічну модель і пропонує новий набір ремінців. По характеристикам Samsung схожий з Apple, з мінімальними відмінностями, але ціна на пристрій залишається приємною і вже як і зазначалось вище це їх ціна.

Тепер перейдемо до бренду Huawei, він на рівні функцій та технологій та такий же як Samsung та Apple які було розглянуто вище, це непоганий розумний годинник який поєднує стильний дизайн який в більшості надихався дизайном механічних годинників. Годинник має автономність до 14 днів без підзарядки, хоч і Huawei була позбавлена Android на кілька років, але бренд все ще залишається еталоном на ринку розумних годинників. Цей годинник виділяється своїм зовнішнім виглядом і восьмикутною рамою. Завдяки новітній системі TruSense годинник може запропонувати ще більш просунуті функції для моніторингу здоров'я та функції для спорту які краще порівняно з моделями минулих років.

Що стосується самого годинника, команда Huawei працювала над його дизайном, щоб зробити його унікальним та запам'ятовувачим.

Завдяки тому що годинник зроблений з титану він має дуже міцний корпус, легко сидить на зап'ясті і має невелику вагу порівняно з іншими годинниками.

Годинник Huawei це особливо стійкий годинник зі склом, захищеним сапфіром. Розумний годинник також як і моделі які розглядались вище працює у воді на глибині до 50 метрів завдяки захисним сертифікатам 5ATM і IP69K.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		15



Рисунок 2.4 – Розумний годинник Huawei

Головною особливістю годинників Huawei є те, що вони працюють під управлінням розробленої компанією Huawei, операційної системи Harmony OS. Бренд з часом зміг удосконалювати свою операційну систему так, щоб зробити її зручною та з багатьма наборами функцій для повсякденного користування.

Інтерфейс Huawei дуже нагадує інтерфейс Apple з плаваючим меню, яке можна збільшувати або зменшувати.

Щоб налаштувати Huawei, вам знадобиться програма Huawei Health, яка доступна на iOS, а також на Android, оскільки годинник сумісний з двома операційками. Це має перевагу, оскільки об'єднує всю інформацію про ваше здоров'я, стан та спорт, а також з параметрами які тісно пов'язані з годинником, наприклад вибір стилю циферблата або сповіщення.

Однак доступу до Google Play Store або App Store немає. Натомість є галерея додатків, яка по суті, обмежена програмами які вже встановлені на годиннику.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		16

Huawei пропонує більш глибокий моніторинг здоров'я та спорту порівняно з попередніми поколіннями годинників з їхнього бренду та завдяки новій системі TruSense, цей пристрій забезпечує кращу точність зібраних даних про ваше здоров'я.

Підсумовуючи, можна сказати, що цей пристрій демонструє інший підхід до побудови смарт-годинника: акцент на високій автономності, спортивних функціях і моніторингу здоров'я і простому дизайну. Цей годинник це приклад пристрою, де роль відіграє зручність щоденного використання та варіативність корисних функцій.

Перейдемо від популярних брендів до маловідомих та які акцентують увагу на одному вектору розвитку, наприклад як Garmin які відомі своїми розумними годинниками для людей з активним способом життя, та виробами для навігації, наприклад годинник від них, став першим пристроєм для компанії поєднуючи досвід обох лінійок інших серій годинників Garmin з новими програмними та апаратними можливостями. Наприклад, тепер годинник має AMOLED-екран, вбудований динамік та мікрофон, а також багато додаткових функцій, які роблять пристрій більш корисним.



Рисунок 2.5 – Розумний годинник Garmin

Взагалом, годинник має наступні головні переваги цього які полягають в серйозному захисті пристрою, та вибору багатьох функцій та можливостей які пов'язані

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						17
Вим	Арк.	№ докум.	Підпис	Дата		

з спортом та з активностями по типу кемпінгу чи хайкінгу. Серед ключових характеристик годинник має: надійний металевий корпус, який захищає від падінь та вологи, 32 ГБ пам'яті, вбудований мікрофон та динамік, голосовий помічник, безліч функцій для професійних занять спортом, тренувань, бігу, функції для екстремального відпочинку, повноцінний GPS та можливість завантажувати карти, ліхтарик, неймовірна автономність.

Можливості годинника, його функціональність та інтерфейс: 32 ГБ вбудованої пам'яті яку можна використовувати для завантаження карт або пісень через вбудований плеєр, що дозволяє відтворювати музику тільки через вбудований динамік.

З корисних функцій є датчик освітлення, фірмовий застосунок для безконтактної оплати, ліхтарик, гіроскоп, барометр та акселерометр повноцінна підтримка геолокації, завдяки власному GPS модулю, з підтримкою різноманітних карт.

Годинник може вести користувача по GPS-треку, для цього тут є компас та висотомір.

Завдяки тому що пристрій оснастили мікрофоном, це дає змогу приймати дзвінки та відповідати на них з годинника де пристрій буде як аналог мікрофону для телефона. Ще є функція голосових команд, яка працює самостійно без прив'язки до телефона, також пристрій має свій месенджер.

Голосом можна давати команди типу запуску певних спортивних режимів, таймера, створення замітки, визначення пульсу тощо.

Нова модель отримала п'ять герметичних кнопок з пружним натисканням. Це залишає корпус максимально захищеним і дозволяє використовувати на великій глибині.

Взаємодія з смартфоном та ПЗ: Garmin підключаються до пристроїв через вбудований модуль Bluetooth та працюють з фірмовим застосунком Garmin Connect.

Взагалом застосунок для годинника Garmin Connect дуже корисний в використанні для відстежування фізичної активності, занять спортом, професійних тренувань тощо. Він має багато налаштувань та можливостей, а ще це має вбудований месенджер як було написано вище. Завдяки зручності застосунка спортсмени можуть домовлятися про певні події, ділитися досягненнями, досвідом, тренуваннями тощо.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						18
Вим	Арк.	№ док.ум.	Підпис	Дата		

Годинник Garmin можна сміливо розглядати як надійний аналог годинника Amazfit який був розглянутий першим але в порівнянні з ним Garmin має більше можливостей та також був розроблений в США що дає йому більше довіри ніж китайському бренду.

Перейдемо до одного з відомих брендів який насправді нещодавно на ринку годинників та інших систем, компанія Google від минулих моделей годинник відрізняється оснащеним широким набором датчиків: компасом, альтиметром, барометром, гіроскопом, акселерометром, пульсометром, датчиком рівня кисню в крові та датчиком температури поверхності тіла.



Рисунок 2.6 - Розумний годинник Google

Разом із годинником компанія представила три нові функції для відстеження фізичних показників: Daily Readiness, Cardio Load та Target Load, які стежать за серцевим ритмом, якістю сну та рівнем активності користувача, також від минулих версій годинника відрізняється кращою автономністю, продуктивністю та розміром в 45мм.

Годинник від Google найкраще розкриває свій потенціал у рамках екосистеми Google, де присутньо багато функцій взаємодії з іншими пристроями від компанії та їх

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		19

програмами такими як: Nest Doorbell, Chromecast, Google Calendar та Google Maps з Google TV.

Також інтеграція з Fitbit стала ще кращою. Окрім звичних даних про пульс, сон тощо, з'явилися алгоритм Readiness для оцінювання рівня відновлення після навантажень, Cardio Load для відстеження навантаження, а в застосунку Fitbit додана нова панель і рекомендовані плани тренувань, що створені на базі ШІ.

Також з головних функцій є виявлення втрати пульсу: у разі екстреної зупинки серця девайс автоматично викликає допомогу та передає ваше місцезнаходження.

Відходячи від іншого популярного бренда ми знову переходимо до незнайомих та бюджетних варіантів на ринку.



Рисунок 2.7 Розумний годинник Ice-Watch

Ice-Watch це бельгійська компанія яка з'явилась нещодавно на нашому ринку, компанія була заснована в 2006 році і з тих пір зарекомендувала як виробник яскравих та доступних годинників. Ключовим покращенням порівняно з попередніми версіями стала інтеграція GPS для точного відстеження маршрутів без підключення до смартфона, що особливо корисно для бігунів та велосипедистів, а також підтримка застосунку Strava. Годинник оснащений акселерометром, шагоміром, пульсометром, гіроскопом та оксиметром, підтримує понад десять

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		20

спортивних режимів, зокрема ходьбу, біг, велоспорт, плавання, теніс, йогу, бокс і лижі.

Звичайно данна модель годинника буде поступатись в функціях наприклад з годинником Huawei але, він буде дешевше та доступніше ніж він.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки системи розумний годинник було обрано мікроконтролер ESP32 DEVKITV1. ESP-32 це мабуть найуніверсальніша серія мікроконтролерів для розробки різних простих або навіть достатньо складних IoT-пристроїв, по багатьом причинам та й з зручності і простоті використання сімейство цих мікроконтролерів, використовують по всьому світу. Причини вибору мікроконтроллера ESP32 DEVKITV1, цей пристрій розробника DEVKITV1 який побудований на мікромодулі ESP-WROOM-32, який має багато корисних функцій таких як вбудований модуль RTC, сам модуль призначений для широкого спектра застосувань, починаючи від мікропотужних мережевих датчиків до найскладніших програм, наприклад, таких як кодування голосу або потокова передача музики та MP3. На модулі зібрана вся необхідна мінімальна периферія, достатня для швидкого і комфортного старту роботи з ESP-WROOM-32.

ESP-WROOM-32 виконаний на базі популярного двоядерного чіпсета ESP32, із змінною тактовою частотою від 80 МГц до 240 МГц, можливістю індивідуального управління і живлення.

Модуль розроблений для переносної і автономної електроніки та додатків інтернет-речей, виконаний в мініатюрному корпусі 25,5 мм x 18 мм, має на борту Flash пам'ять, кварц 40 МГц і PCB антену, що забезпечує відмінні RF характеристики.

ESP-WROOM-32 має багату периферію, що включає в себе такі інтерфейси як UART, SPI, I2C, I2S, роз'єм для SD карти, інфрачервоний порт, інтерфейс для підключення ємнісної сенсорної панелі.

Однією з особливостей модуля є наднизьким споживання і гнучкий вибір «сплячих» режимів, що дозволяють отримати цифри до 20мкА (deep sleep mode).

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						21
Вим	Арк.	№ докум.	Підпис	Дата		

Модуль підтримує весь стек протоколів стандартів WiFi 802.11n і BT4.2, забезпечуючи даний функціонал через інтерфейси SPI/SDIO або I2C/UART.

ESP32 – це серія недорогих і малопотужних чіпів, розроблених компанією Espressif Systems, які є потужнішим наступником ESP8266. ESP32 був випущений у 2016 році і має двоядерний дизайн з підтримкою WiFi і дворежимного Bluetooth. Мікроконтролер оснащений 32-бітовим мікропроцесором Tensilica Xtensa LX6, що працює на частоті 240 МГц, і співпроцесором з ультранизьким енергоспоживанням

Модуль, плата ESP 32 – це покращена версія плати ESP 8266, яка поєднує WiFi та Bluetooth передавачі, у такий спосіб роблячи плату **ESP32 DEVKIT** одну з найпопулярніших у проєктах програмування розумного дому та інших розумних пристроїв.

З огляду на той факт, що плата може програмуватися з використанням програми Arduino IDE, вона є еталоном для початкових стартапів. Плата ESP32 обладнана двоядерним 32-бітовим мікропроцесором LX6 Tensilica Xtensa, що працює на частоті 160 або 240 МГц, водночас поєднує в собі 30 повноцінних виводів.



Рисунок 2.8 – Мікроконтролер ESP32 DEVKITV1

Відмінні особливості мікроконтролера ESP32 DEVKITV1:

- Повна інтеграція мінімальної обв'язки;
- Просте програмування модуля через USB;

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		22

- Підтримка dual mode Bluetooth: «classic» і BLE;
- Швидкість Wi-Fi: 802.11 b/g/n до 150 Мбіт/с;
- Підтримка режимів Wi-Fi: клієнт, точка доступу, Sniffer, Wi-Fi Direct;
- Мінімальна чутливість -98 dBm;
- Енергоспоживання до 20мкА (deep sleep mode);
- Оновлення ПО по повітрю;
- Можливість підключення 4 x 16MB зовнішньої QSPI Flash і SRAM;

Технічні характеристики мікроконтролера ESP32 DEVKITV1:

- USB-UART конвертер: CP2102;
- Напруга живлення: 5В;
- Максимальний струм стабілізатора напруги: 800мА;
- Wi-Fi Стандарти: FCC/CE/IC/TELEC/KCC/SRRC/NCC;
- Протоколи: 802.11 b/g/n/d/e/i/k/r (802.11n до 150 Мбіт/с);
- A-MPDU і A-MSDU підтримка та підтримка захисного інтервалу в 0.4 сек;
- Частотний діапазон: ГГц 2.4 ~ 2.5;
- Bluetooth Протоколи: Bluetooth v4.2 BR/EDR і BLE specification;
- Радіо NZIF приймач з чутливістю: -98 dBm;
- Передавач: Class-1, class-2 і class-3 AFH;
- Аудіо: CVSD і SBC;
- Апаратні засоби та інтерфейси: SD, UART, SPI, SDIO, I2C, LED PWM, MotorPWM, I2S, IR;

- GPIO, сенсорний датчик, ADC, DAC, LNA передпідсилювач;
- Датчики «на борту»: Hall sensor, температурний датчик ;
- Генератори: кварцовий 26 МГц і 32 кГц;
- Живлення мікромодуля: В 2.2 ~ 3.6;
- Робочий струм, середній: 80 мА;
- Робочий струм піковий: 500 мА;
- Діапазон робочих температур: -40 ... +85 °С;
- Відстань між контактними пінами: 25.5 мм;

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		23

Структурна схема ESP32

На зображенні 2.9 показана функціональна блок-схема мікроконтролера ESP32 який відомий та популярний Wi-Fi та Bluetooth модуль, який часто використовується людьми в IoT пристроях, smart годинниках та різних системах і Arduino проєктах.

Основні компоненти ESP32:

Radio

Відповідає за бездротовий зв'язок:

- RF receive - прийом радіосигналу;
- RF transmit - передача сигналу;
- Clock generator - генератор тактової частоти;
- Balun та Switch - узгодження й перемикання антени;

Bluetooth та Wi-Fi

Окремі модулі для роботи мереж:

- Bluetooth baseband - базова обробка Bluetooth;
- Bluetooth link controller - керування Bluetooth-з'єднанням;
- Wi-Fi baseband - обробка Wi-Fi сигналів;
- Wi-Fi MAC - керування мережевими пакетами Wi-Fi;

Cryptographic hardware acceleration

ESP32 має вбудовані засоби захисту даних:

- RSA - алгоритм асиметричного шифрування;
- AES - швидке симетричне шифрування;
- SHA - хешування даних;
- RNG - генератор випадкових чисел.

Core and memory

Основна обчислювальна частина:

- Xtensa LX6 - 32-бітний процесор (1 або 2 ядра);
- ROM - постійна пам'ять;
- SRAM - оперативна пам'ять;

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						24
Вим	Арк.	№ докум.	Підпис	Дата		

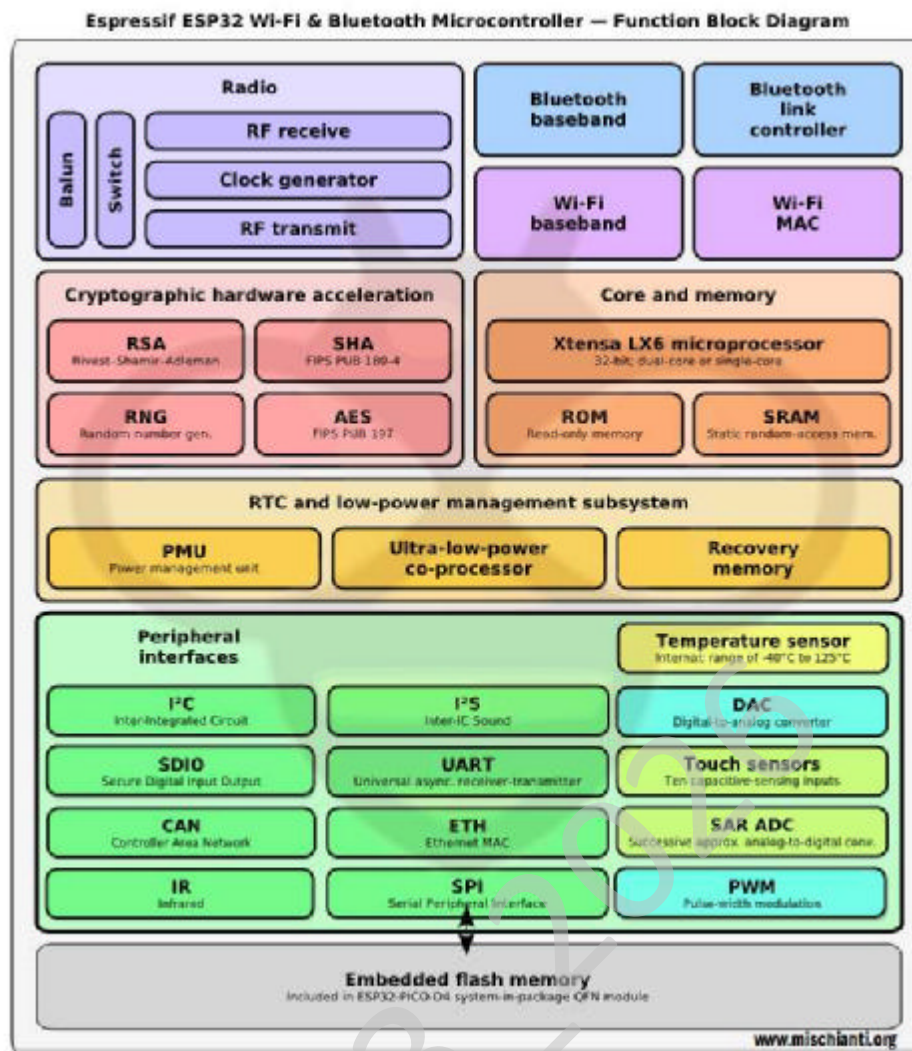


Рисунок 2.9 - Структурна схема внутрішніх компонентів мікроконтролера

RTC and low-power management subsystem

Забезпечує низьке енергоспоживання:

- PMU - модуль керування живленням;
- Ultra-low-power co-processor - ультраекономний співпроцесор;
- Recovery memory - пам'ять відновлення.

Peripheral interfaces

ESP32 підтримує багато способів підключення датчиків та модулів:

- I2C - підключення сенсорів;
- SPI - швидкі периферійні пристрої;
- UART - обмін даними;

Вим	Арк.	№ докум.	Підпис	Дата

ВКРБ-122.26.0016.00.00.ПЗ

Арк.

25

- IS - аудіоінтерфейс;
- SDIO - робота з SD-картами;
- CAN - автомобільні мережі;
- ETH - Ethernet;
- IR - інфрачервоний зв'язок.

Аналогові та цифрові модулі

Додаткові функції:

- ADC - аналого-цифровий перетворювач;
- DAC - цифро-аналоговий перетворювач;
- PWM - широтно-імпульсна модуляція;
- Touch sensors - сенсорні входи;
- Temperature sensor - внутрішній датчик температури.

Embedded flash memory

Призначена для:

- Зберігання прошивки;
- Даних програми;
- Конфігурацій пристрою;

OLED LCD 1.3 дюйма – Компактний графічний OLED-дисплей 1.3 дюйма на контролері SSH1106 для мікроконтролерів Arduino, MSP430, ESP8266/ESP32, AVR, PIC, STM32 та інших. Завдяки OLED-технології світлодіоди випромінюють світло без додаткового підсвічування. Дисплей має високу якість відображення та швидкість з'єднання, широкий кут огляду, мінімальне енергоспоживання.

Дисплей має наступні характеристики, SPI інтерфейс (I2C при розпайці резисторів R1, R5) Діапазон робочих температур від -40°C - 70°C . Має 7 пінів, та контролер SSH1106. Мікромодуль працює в діапазоні напруги від 3.3-5В. Має наступну роздільну здатність 128*64. Розмір 35 x 33мм. Також є можливість вибору сенсорного дисплею.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						26
Вим	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.10 - OLED LCD дисплей на 1.3 дюйма

TP4057 Type C для Li-ion LiPo 1S 1A – модуль зарядки та розрядки на мікросхемі TP4057 призначений для роботи з однією літійовою батареєю типу Li ion або LiPo формату 1S з номінальною напругою 3.7 В. Оснащений роз'ємом Type C та вбудованими схемами захисту від перезаряду, перерозряду і перегріву. Підтримує алгоритм зарядки CC CV та забезпечує стабільний контроль процесів заряджання і розрядки.

Li-ion 302530-200 x 200mAh – звичайна батарея для пристрою.

Елементи корпусу та кнопки – потрібні для того щоб вмикати та вимикати пристрій та взаємодіяти з ним.

Для данного проєкту була обрана мова програмування C/C++, тому що вона сумісна з мікроконтролером. ESP-32 DEVKITV1 підтримує мову програмування C/C++, яка є стандартом для вбудованих систем, включаючи мікроконтролери.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						27
Вим	Арк.	№ докум.	Підпис	Дата		

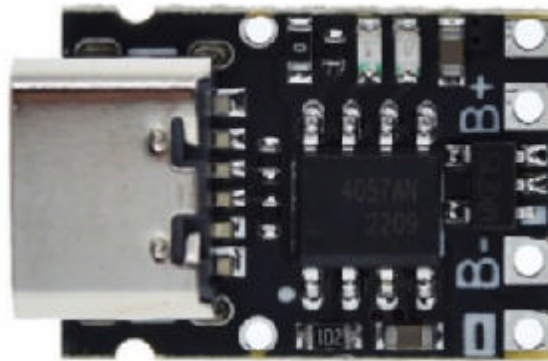


Рисунок 2.11 - Модуль зарядки та розрядки TP4057 Type C

Завдяки цьому, є можливість працювати з апаратурними ресурсами на низькому рівні, що дозволяє оптимізувати роботу з ESP32 та отримувати високу ефективність програм.

Середовищем розробки стала Arduino IDE. Arduino IDE – це ПЗ, призначене для створення власних програм для платформи Arduino, а також для інших мікроконтролерів. Це популярне середовище розробки, особливо корисне для творчих конструкторів, що використовують Arduino для створення простих систем автоматизації та робототехніки. Середовище надає велику кількість вбудованих бібліотек, що полегшують розробку програм. Їх можна знайти в офіційному репозиторії Arduino IDE, а також на спеціалізованих вебсайтах та форумах, де користувачі діляться своїми власними розробками та бібліотеками для використання в проектах. Одним з найбільших таких сайтів є Github, на якому є майже всі бібліотеки які тільки треба для будь-яких проєктів, також бібліотеки можна знайти на сайтах, де продають датчики, модулі і тд. Враховуючи вищесказане, мова програмування C/C++ та Arduino IDE – це ідеальний варіант для реалізації будь-яких проєктів які тільки можна придумати.

Вим	Арк.	№ докum	Підпис	Дата

ВКРБ-122.26.0016.00.00.ПЗ

Арк.

28

Також для данного проєкту була розроблена локальна програма симулятор на мові C++, з тими ж самим функціями як і реальний прототип, для того щоб можна було продемонструвати можливості системи розумний годинник без наявності реального прототипу при собі.

Для розробки Android застосунку для автентифікації було обрано середовище Kotlin. Kotlin – це мова програмування що працює поверх JVM та також компілюється в JavaScript. Автори ставили перед собою мету створити лаконічнішу та типобезпечнішу мову, ніж Java. Наслідками спрощення, порівняно зі Scala стали також швидша компіляція та краща підтримка IDE. Kotlin послаблює обмеження Java, що дозволяє статичним методам та змінним існувати лише в тілі класу. Статичні об'єкти та функції можуть бути визначені на верхньому рівні пакету без необхідності створення надлишкового рівня класів. Для сумісності з Java у Kotlin передбачено анотацію `JvmName`, яка визначає ім'я класу, що використовується, коли пакунок переглядається з проєкту Java.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для інтелектуальної системи розумний годинник з автентифікацією даних через мобільний додаток.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

- Визначити призначення системи та область її застосування.
- Провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках.
- Обґрунтування вибору засобів для побудови системи, таких як мікроконтролер, локальний симулятор - демонстрація, мови програмування.
- Написання ПЗ розумний годинник з автентифікацією даних через мобільний додаток.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		29

- Протестувати систему, щоб виявити та усунути можливі помилки та недоліки.
- Написати перелік умовних позначень, символів, одиниць та термінів, що будуть використані при написанні роботи.
- Надати список літератури, що використалась при написанні роботи.
- Написати анотацію на українській та англійській мовах.

К6ПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						30
Вим	Арк.	№ докум.	Підпис	Дата		

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розроблена система «розумний годинник» складається з двох взаємопов'язаних та основних програмних компонентів: основної програми годинника та мобільного додатку. Взаємодія між якими побудована на базі локального HTTP, який дозволяє реалізувати наступні цикли керування, аутентифікації, синхронізації часу та встановлення будильника та інших.

Розроблене ПЗ передбачає два способи функціонування основного пристрою. Перший – фізичний, на якому ПЗ розгорнуте безпосередньо на мікроконтролері ESP32 DEVKITV1. У цьому вигляді ESP32 представляє повноцінний пристрій годинника: він обробляє HTTP запити від мобільного додатку автентифікатора, відображає стан та інформацію на дисплеї.

Другий, демонстраційний, при якому ПЗ написане на C++ за допомогою Visual Studio та запускається на ПК та відтворює всі функції годинника, без потреби у фізичному пристрої. Обидва способи функціонування системи розумний годинник реалізують однаковий REST API та логіку поведінки, з якими мобільний застосунок взаємодіє з обома без змін у своєму коді.

Подібна архітектура систем є характерною для IoT-систем в вигляді пристрій плюс мобільний додаток. Зокрема, ESP32 широко застосовується для побудови різних розумних пристроїв, де мікроконтролер виступає як вбудований HTTP сервер, а мобільний застосунок як клієнт. Такий підхід дозволяє відокремити логіку пристрою від інтерфейсу, а також набагато спрощує тестування та полегшує подальше масштабування. Архітектура ESP32 з незалежним керуванням ядрами дозволяє ефективно розподіляти завдання, в той час як інші вбудовані модулі в мікроконтролер можуть полегшувати прошивку пристрою.

Функціонально розумний годинник реалізує три групи функцій: інформаційну, керувальну та демонстраційну. Інформаційна функція забезпечує відображення

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						31
Вим	Арк.	№ докум.	Підпис	Дата		

поточного часу, дати, дня тижня, кількості пройдених кроків, рівня заряду акумулятора в пристрої, температури, стан будильника та статус автентифікації. Керувальна функція відповідає за синхронізацію часу, встановлення будильника, блокування та розблокування пристрою через додаток. Демонстраційна функція дозволяє повноцінно продемонструвати роботу системи на комп'ютері без наявності фізичного пристрою.

Автентифікація є головним завданням бакалаврської роботи та також може виступити як механізм безпеки для системи розумний годинник. Автентифікація реалізована на основі PIN коду: користувач вводить чотиризначний PIN у мобільному додатку, після чого застосунок надсилає запит до годинника або ПК в симуляторі. Сервер перевіряє PIN та повертає відповідь зі статусом автентифікації. У разі успішної автентифікації мобільний застосунок отримує доступ до пристрою. У разі невідповідності PIN коду сервер повертає код відповіді HTTP 401, а застосунок відображає повідомлення про помилку. Вищесказаний підхід до автентифікації системи розумний годинник, із застосуванням PIN кодів є типовим рішенням для локальних IoT систем, де пристрій і мобільний застосунок знаходяться в одній мережі.

Мобільний застосунок був розроблявся мовою Kotlin для Android. Він складається з двох основних екранів: екрана входу (LoginActivity) та панелі керування (DashboardActivity). На екрані входу користувач вибирає мову інтерфейсу (українська або англійська) і вводить PIN код для підключення до годинника. Після успішної автентифікації відкривається панель керування, яка відображає актуальний стан пристрою та дозволяє користуватися панеллю керування. Весь обмін даними між мобільним застосунком і годинником здійснюється через HTTP-клієнт (WatchApiClient), реалізований на базі класу HttpURLConnection без використання сторонніх мережевих бібліотек.

Демонстраційна функція пристрою системи, написана мовою C++, запускається на ПК як локальний HTTP сервер і реалізує набір ендпоінтів REST API, як і на фізичний прошивці ESP32. Такий підхід апаратної поведінки ПЗ дозволяє повноцінно демонструвати та взаємодіяти мобільним додатком з пристроєм без необхідності мати фізичний прошитий мікроконтроллер з дисплеєм. Це дуже цінно в навчальних цілях де студенти зможуть вивчати базові принципи IoT, локальних API та Android розробку, за

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						32
Вим	Арк.	№ докум.	Підпис	Дата		

допомогою звичайного ПК без потреби в мікроконтролері.

Цикл взаємодії системи в обох функціональних варіантах виглядає наступним чином:

- запуск прошитого мікроконтролера або Демонстрації на ПК.
- відображення стану системи.
- введення PIN у мобільному додатку.
- автентифікація через HTTP /auth.
- отримання даних стану через /status.
- виконання керувальницьких дій (синхронізація часу, будильник, блокування тощо) через відповідні ендпоінти.
- відображення оновленого стану на дисплеї годинника та мобільному додатку.

Обрана архітектура системи є модульною: логіка пристрою, мобільний інтерфейс користувача та firmware-версія для ESP32 розділені на окремі частини. Такий підхід дозволяє незалежно розробляти та перевіряти кожний компонент, не порушуючи роботу всієї системи. Мобільний застосунок відповідає за взаємодію з користувачем, введення PIN-коду та відображення стану годинника. Симулятор або прошивка ESP32 відповідає за збереження стану пристрою, перевірку автентифікації та виконання команд керування.

Використання локального HTTP як основного каналу обміну даними обґрунтоване простотою демонстрації системи розумний годинник. На відміну від BLE, HTTP-запити легко перевірити вручну, аналізуючи налагодження та запускати в середовищі демонстрацію. Для дипломного прототипу важливо, щоб система стабільно працювала під час захисту, тому основну увагу приділили надійному локальному обміні даними, а не роботі з радіоканалами.

Модель взаємодії побудована за принципом клієнт-сервер. У ролі клієнта виступає застосунок, а в ролі сервера годинник або його локальний симулятор. Такий підхід є типовим для розумних систем, у яких фізичний пристрій має власний стан і надає зовнішньому застосунку обмежений набір операцій через API. Завдяки цьому мобільний застосунок не має прямого доступу до внутрішньої пам'яті пристрою, а працює лише через визначені ендпоінти.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		33

Для представлення стану годинника використовується єдина логічна модель даних. Вона включає поточний час, дату, день тижня, рівень заряду батареї, кількість кроків, температуру, час будильника, поточний екран та статус автентифікації. У симуляторі ця модель реалізована у вигляді класу WatchState, у firmware-версії – у вигляді структури WatchState, а в Android-застосунку – у вигляді data class WatchStatus. Завдяки цьому всі компоненти системи працюють з однаковими поняттями.

Сенсорні дані у демонстраційному режимі генеруються програмно. Це стосується кількості кроків, температури та заряду батареї. Такий підхід дозволяє показати динамічну поведінку пристрою без підключення реальних датчиків. У майбутній апаратній версії ці програмні значення можуть бути замінені реальними показниками акселерометра, температурного датчика та контролера батареї без зміни логіки мобільного застосунку.

Автентифікація реалізована як окремий етап взаємодії з пристроєм. До проходження автентифікації користувач може отримувати інформаційний стан годинника, але не може змінювати критичні параметри, такі як час або будильник. Це дозволяє продемонструвати базову модель розмежування доступу: читання стану є відкритою дією, а керування пристроєм вимагає підтвердження користувача.

У системі передбачено локалізацію інтерфейсу українською та англійською мовами. Це реалізовано як у мобільному застосунку, так і у web UI годинника. Наявність локалізації підвищує зручність використання системи та відповідає вимогам дипломної роботи, оскільки демонстрація може проводитися українською мовою, а технічні назви компонентів можуть залишатися зрозумілими в англomовному контексті.

3.2 Розробка структурної схеми

Структурна схема системи «розумний годинник» відображає склад та взаємозв'язки між трьома основними підсистемами: мобільним застосунком Android, локальним симулятором watch_sim та прошивкою ESP32. Схему наведено на рисунку 3.1.

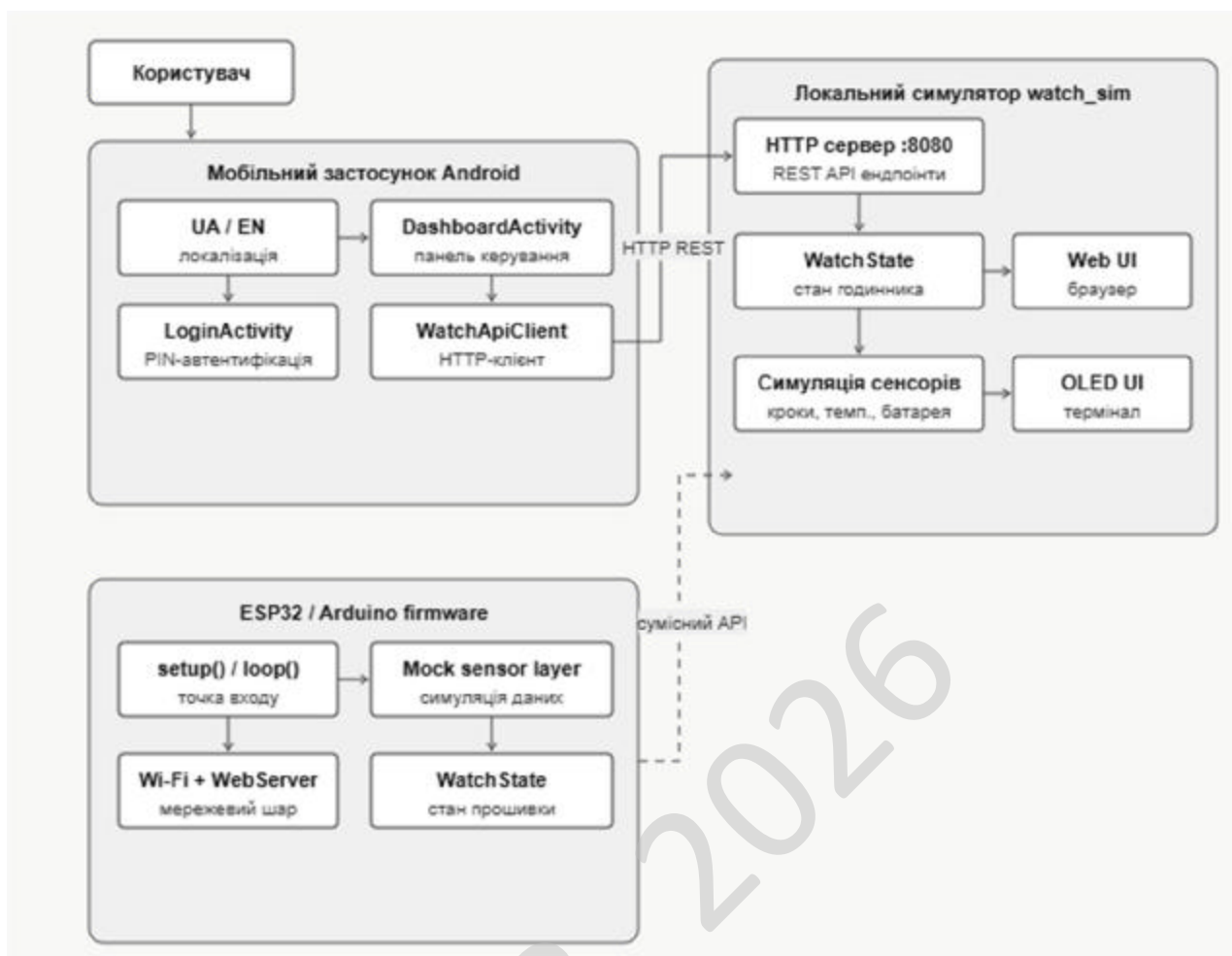


Рисунок 3.1 – Структурна схема системи

Перша підсистема – мобільний застосунок Android – містить чотири компоненти. Модуль UA/EN локалізації забезпечує вибір мови інтерфейсу за допомогою класу LocaleHelper; вибрана мова зберігається між сеансами та застосовується при перезапуску будь-якої активності. LoginActivity реалізує екран PIN-автентифікації: користувач вводить чотиризначний код, після чого застосунок формує POST-запит і очікує відповіді сервера. DashboardActivity є головною панеллю керування – вона відображає актуальний стан годинника та надає елементи управління для синхронізації часу, встановлення будильника та блокування пристрою. WatchApiClient виступає єдиним HTTP-клієнтом застосунку: він виконує всі GET- та POST-запити до сервера годинника і повертає результати у вигляді об'єктів типу Result.

Друга підсистема – локальний симулятор `watch_sim` є програмним еквівалентом фізичного годинника та містить п'ять блоків. HTTP сервер :8080 приймає вхідні запити від мобільного застосунку та маршрутизує їх до відповідних обробників REST API (ендпоінти `/status`, `/auth`, `/sync-time`, `/alarm`, `/lock`). `WatchState` зберігає поточний стан системи: час, дату, день тижня, кількість кроків, рівень заряду, температуру, час будильника та статус автентифікації. Блок симуляції сенсорів генерує реалістичні змінювані значення кроків, температури та заряду батареї у фоновому режимі. Web UI забезпечує доступ до стану годинника через браузер, а OLED UI відтворює термінальний інтерфейс – текстовий аналог дисплея фізичного пристрою.

Третя підсистема – прошивка ESP32 / Arduino firmware – відтворює ту саму логіку роботи на реальному апаратному пристрої та містить чотири блоки. Функції `setup()` та `loop()` є точкою входу прошивки: `setup()` виконує одноразову ініціалізацію всіх модулів, `loop()` забезпечує безперервне опитування HTTP-запитів і оновлення стану. Блок Wi-Fi + WebServer відповідає за підключення ESP32 до бездротової мережі та запуск вбудованого HTTP-сервера на порті 80. Mock sensor layer імітує показники сенсорів, аналогічно до симулятора. `WatchState` firmware є структурою даних прошивки, що зберігає поточний стан пристрою в оперативній пам'яті мікроконтролера.

Між підсистемами існують два типи зв'язків. Суцільна стрілка між `WatchApiClient` та HTTP сервером :8080 позначає основний комунікаційний канал – локальний HTTP REST API. У середовищі Android-емулятора адреса сервера має вигляд `10.0.2.2:8080`; при роботі з фізичним ESP32 використовується IP-адреса пристрою в локальній Wi-Fi-мережі. Пунктирна стрілка між підсистемами ESP32 та симулятора позначає сумісність API: обидві підсистеми реалізують ідентичний набір ендпоінтів, тому мобільний застосунок взаємодіє з ними без жодних змін у своєму коді. Це дозволяє вільно перемикатися між режимом демонстрації на ПК та роботою з реальним пристроєм.

Структурна схема демонструє, що система не є монолітною програмою, а складається з кількох взаємопов'язаних рівнів. Перший рівень: користувацький, представлений мобільним застосунком. Другий рівень – пристрій або його програмний еквівалент, який приймає запити, зберігає стан і виконує команди.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		36

Третій рівень: *firmware*-представлення, що показує можливість перенесення логіки на фізичний ESP32-пристрій.

Компонент WatchApiClient на структурній схемі винесено окремо, оскільки він є єдиною точкою мережевої взаємодії Android-застосунку з годинником. Саме через нього виконуються запити GET /status, POST /auth, POST /sync-time, POST /alarm та POST /lock. Така централізація спрощує супровід коду: у разі зміни адреси сервера або формату відповіді достатньо змінити один модуль, а не всі екрани застосунку.

У блоці watch_sim важливу роль відіграє WatchState. Він є центральним сховищем стану системи. HTTP-сервер не зберігає дані самостійно, а лише приймає запити та звертається до WatchState для читання або зміни інформації. Це дозволяє чітко розділити мережеву логіку та логіку стану пристрою.

Web UI та Terminal OLED UI на структурній схемі є двома різними способами відображення одного й того самого стану. Web UI використовується для наочного браузерного представлення годинника, а Terminal OLED UI потрібен для швидкої демонстрації без браузера. Обидва інтерфейси не створюють власних даних, а лише відображають інформацію зі спільного стану пристрою.

Firmware-блок на схемі підтверджує, що програмна архітектура підготовлена до роботи на реальному пристрої. У ньому присутні setup() і loop(), блок Wi-Fi + WebServer, *firmware*-структура WatchState та mock sensor layer. Ці елементи відповідають типовій структурі Arduino/ESP32-проекту, де ініціалізація виконується один раз, а основна логіка працює у безперервному циклі.

Пунктирний зв'язок між *firmware*-версією та локальним симулятором означає функціональну еквівалентність. Обидва варіанти мають однакову модель стану та однаковий набір API-операцій. Завдяки цьому мобільний застосунок може працювати з демонстраційним сервером на ПК або з фізичним ESP32-пристроєм без зміни основної логіки.

Якщо розглядати систему з позиції майбутнього апаратного пристрою, то ESP32 виступає центральним контролером. До нього можуть бути підключені дисплей, кнопки, акселерометр, температурний датчик та модуль контролю батареї.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						37
Вим	Арк.	№ докум.	Підпис	Дата		

У поточній програмній реалізації ці апаратні блоки замінені web UI, клавіатурним керуванням і tosch-сенсорами, що дозволяє зберегти архітектуру без обов’язкової наявності hardware.

3.3 Розробка функціональної схеми

Функціональна схема системи описує послідовність виконання основних функцій та потоки даних між компонентами під час типового сеансу роботи користувача з системою.

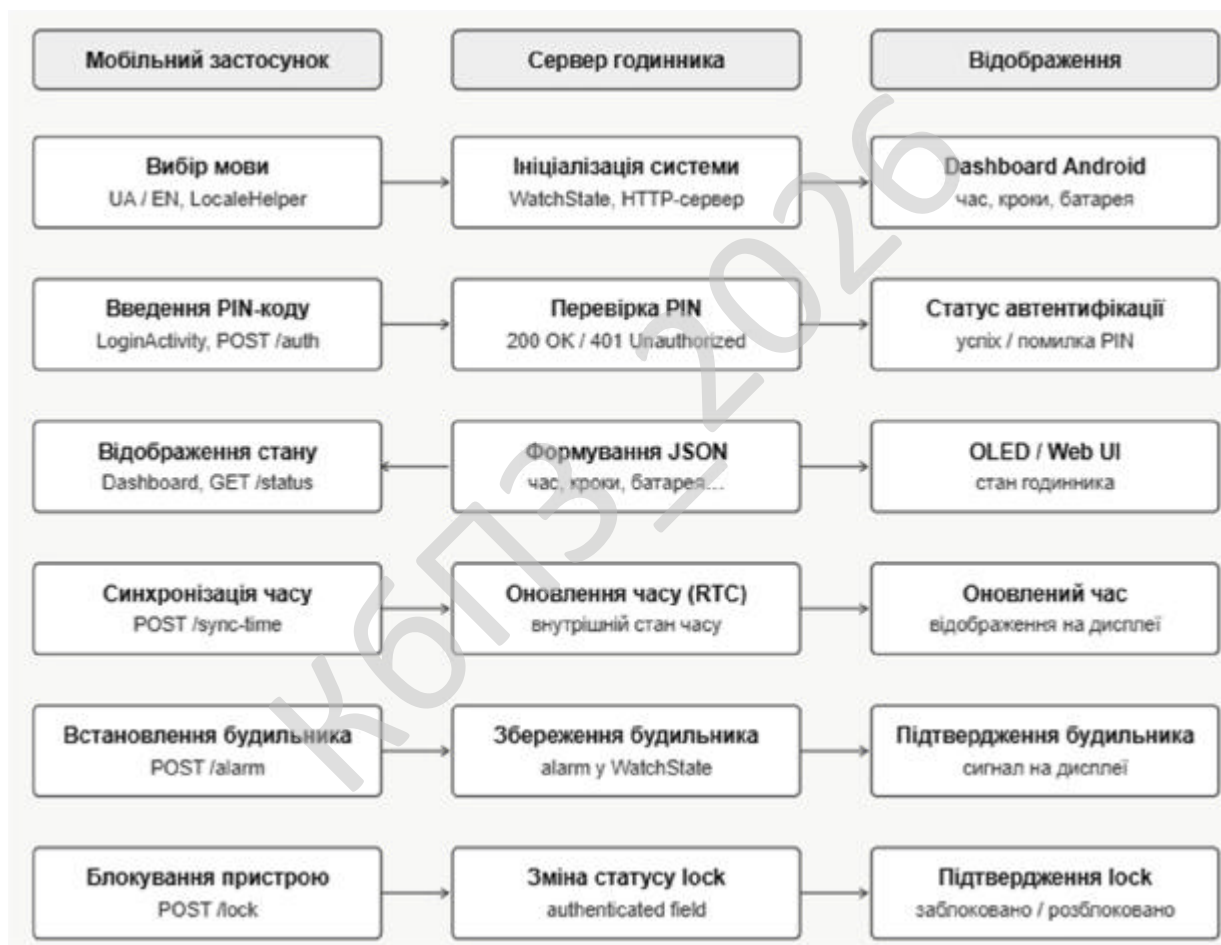


Рисунок 3.2 – Функціональна схема системи

Початковий стан системи, годинник (або симулятор) запущено та очікує вхідних HTTP-запитів. Дисплей відображає поточний час, дату та базові показники у заблокованому режимі. Статус автентифікації - false.

Перший функціональний блок – ініціалізація та запуск. При запуску прошивки ESP32 або симулятора виконуються наступні дії: ініціалізація OLED-дисплею, підключення до Wi-Fi-мережі (у фізичному режимі) або прив'язка до локального порту (у режимі симулятора), ініціалізація RTC-модуля та синхронізація внутрішнього часу, ініціалізація симульованих показників (кроки, температура, заряд батареї), запуск HTTP-сервера та перехід в режим очікування запитів.

Другий функціональний блок – автентифікація. Користувач відкриває мобільний застосунок і вводить чотиризначний PIN-код на екрані LoginActivity. Застосунок формує JSON-об'єкт {"pin": "XXXX"} та надсилає POST-запит до ендпоінту /auth. Сервер годинника перевіряє отриманий PIN із збереженим демонстраційним значенням. У разі збігу сервер повертає {"authenticated": true} зі статусом HTTP 200 та оновлює внутрішній стан автентифікації. У разі невідповідності повертається {"authenticated": false, "error": "Invalid PIN"} зі статусом HTTP 401. На стороні застосунку у разі успіху виконується перехід до DashboardActivity, у разі помилки – відображається відповідне повідомлення.

Третій функціональний блок – отримання та відображення стану. DashboardActivity після завантаження автоматично виконує GET-запит до /status. Отриманий JSON-об'єкт десеріалізується у модель даних WatchStatus, після чого всі поля відображаються на панелі керування: час, дата, день тижня, кроки, заряд батареї, температура, будильник та статус автентифікації. Оновлення стану може ініціюватися як автоматично, так і вручну користувачем.

Четвертий функціональний блок – керувальні дії. Включає три підфункції:

Синхронізація часу – користувач вибирає поточний час у застосунку, застосунок надсилає POST /sync-time з полем {"time": "HH:MM"}, сервер оновлює внутрішній RTC-час та відображає нове значення на дисплеї;

Встановлення будильника – аналогічно синхронізації, POST /alarm з полем {"time": "HH:MM"}, сервер зберігає значення будильника та відображає його на дисплеї;

Блокування/розблокування – POST /lock з полем {"locked": "true/false"}, серверзмінює стан блокування та повертає оновлений статус автентифікації.

П'ятий функціональний блок – завершення сеансу. Після завершення роботи користувач закриває застосунок. Стан годинника зберігається; дисплей продовжує

відображати поточний час. Статус автентифікації зберігається до наступного перезапуску пристрою або явного виклику функції блокування.

Паралельно з обробкою HTTP-запитів симулятор або прошивка виконують фонове оновлення симульованих показників: кроки збільшуються з плином часу, температура та заряд батареї змінюються в межах реалістичних діапазонів. Це забезпечує відображення динамічного стану пристрою при кожному виклику /status.

Функціональна схема показує не лише перелік можливостей системи, а й логіку доступу до них. Усі функції можна умовно поділити на інформаційні та керувальні. Інформаційні функції доступні без автентифікації та включають перегляд часу, дати, батареї, кроків, температури та будильника. Керувальні функції змінюють стан пристрою, тому потребують попереднього введення правильного PIN-коду.

Функція отримання стану реалізована через ендпоінт /status. Вона є базовою для всієї системи, оскільки саме її використовує DashboardActivity для заповнення полів інтерфейсу, а browser UI – для періодичного оновлення екрана годинника. Відповідь /status має формат JSON і містить усі основні поля стану пристрою.

Функція синхронізації часу реалізована через /sync-time. У демонстраційному режимі мобільний застосунок бере поточний час телефону та передає його годиннику. На стороні симулятора час не просто записується як статичне значення, а враховується через програмний зсув від системного часу. Це дозволяє годиннику продовжувати відлік після синхронізації.

Функція встановлення будильника реалізована через /alarm. Перед відправкою запиту Android-застосунок перевіряє формат часу HH:mm, а сервер додатково виконує власну перевірку.

Подвійна перевірка є корисною, оскільки клієнтська валідація покращує зручність користувача, а серверна гарантує коректність даних незалежно від клієнта. Функція блокування реалізована через /lock. Вона дозволяє повернути систему у стан, коли для подальшого керування знову потрібно пройти автентифікацію.

Це важливо для демонстрації повного циклу безпеки: пристрій заблокований, користувач автентифікується, виконує дії, після чого знову блокує пристрій.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		40

Окремою функцією є перемикання екранів годинника. У симуляторі та firmware передбачені екрани Main, Steps, Battery та Alarm. Це дає змогу показати, що годинник має не лише набір API-полів, а й власну логіку відображення інформації, подібну до реального носимого пристрою.

Фонові симуляція даних працює незалежно від дій користувача. Кроки збільшуються через певні інтервали часу, температура змінюється в заданому діапазоні, а заряд батареї поступово зменшується. Завдяки цьому при кожному оновленні /status користувач може бачити, що система має живий стан, а не повертає фіксовані тестові значення.

Функціональна схема також враховує локалізацію. Перемикач UA/EN у мобільному застосунку та web UI не впливає на API, але змінює спосіб відображення даних користувачу. Це правильне розділення відповідальностей: backend повертає структуровані дані, а інтерфейс сам вирішує, якою мовою їх показувати.

3.4 Розробка діаграми процесів

Діаграма процесів описує послідовність дій користувача та системи у ключових сценаріях взаємодії мобільного застосунку з розумним годинником. Основним сценарієм є автентифікація та виконання захищеної операції. До автентифікації користувач може переглядати стан годинника, однак не має можливості змінювати критичні параметри – час або будильник.

Цикл взаємодії починається із запуску системи. Симулятор або прошивка ESP32 ініціалізують структуру даних WatchState (час, кількість кроків, рівень батареї, PIN-код)

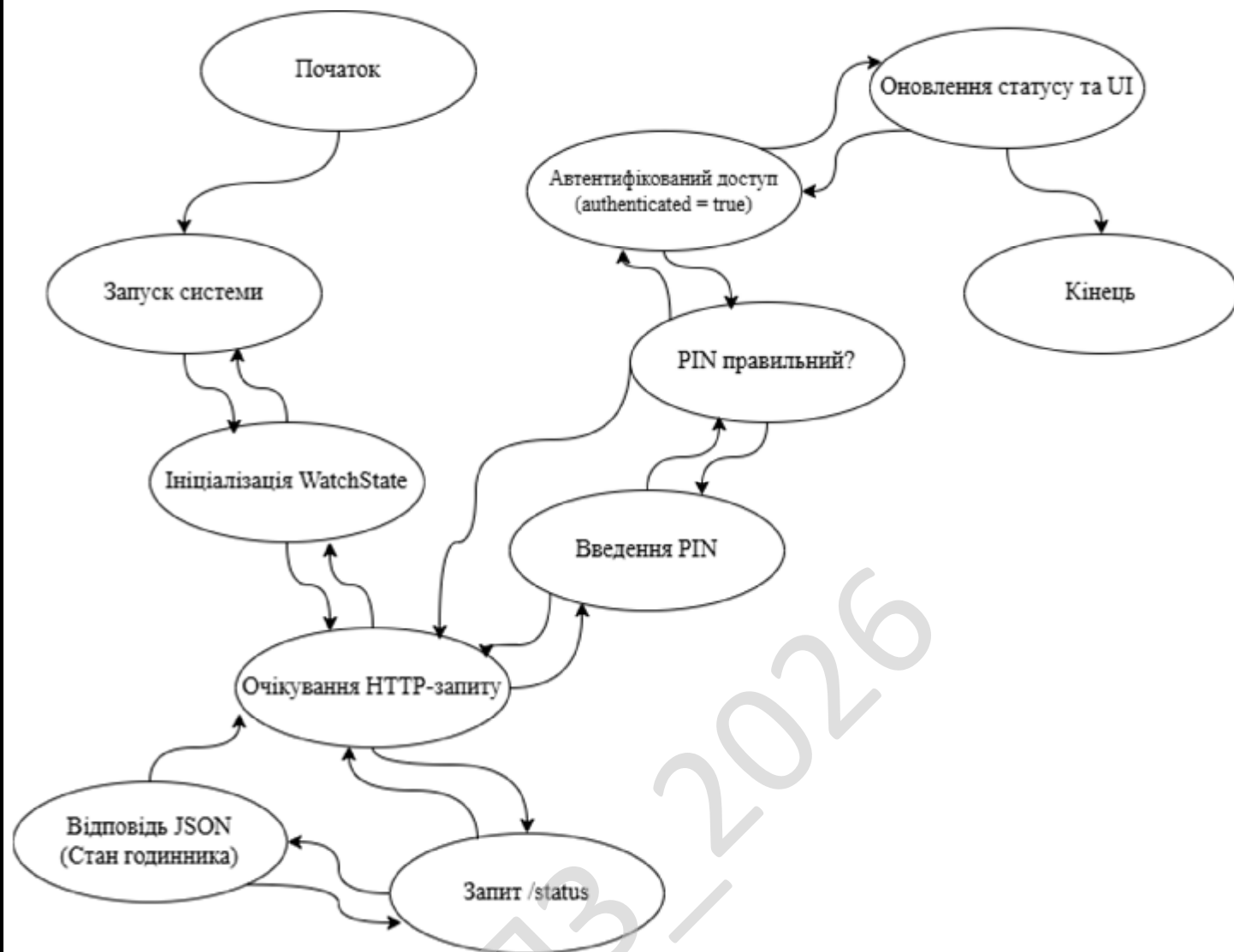


Рисунок 3.3 – Діаграма взаємодії процесів

та запускають HTTP-сервер, після чого система переходить у режим очікування вхідних HTTP-запитів від мобільного застосунку.

Перший сценарій – перегляд стану без автентифікації. Коли Android-застосунок надсилає GET-запит до ендпоінту /status, сервер негайно повертає поточний JSON-стан годинника: час, дату, день тижня, кроки, батарею, температуру, будильник та статус автентифікації.

Цей сценарій не потребує PIN-коду та виконується циклічно під час роботи застосунку.

Другий сценарій – автентифікація. Коли користувач вводить чотиризначний PIN у застосунку, той надсилає POST-запит до ендпоінту /auth. Сервер перевіряє отриманий PIN. Якщо PIN неправильний, сервер повертає HTTP-статус 401 та JSON-відповідь {"authenticated": false, "error": "Invalid PIN"}; стан системи залишається заблокованим. Якщо PIN правильний, сервер повертає {"authenticated": true} зі статусом HTTP 200 та переводить систему у стан автентифікованого доступу.

Третій сценарій: виконання керувальних дій. Цей сценарій доступний лише після успішної автентифікації. Застосунок надсилає POST-запити до одного з трьох ендпоінтів.

- /sync-time – синхронізація часу: {"time": "HH:MM"};
- /alarm – встановлення будильника: {"time": "HH:MM"};
- /lock – блокування або розблокування пристрою: {"locked": "true/false"};

Кожен із цих запитів проходить однакову послідовність обробки на сервері: прийом запиту слідом перевірка вхідних параметрів далі перевірка статусу автентифікації після того оновлення відповідного поля WatchState після формується JSON-відповідь далі оновлення відображення на OLED-дисплеї та панелі Dashboard мобільного застосунку. Такий уніфікований процес обробки зручний для тестування: кожен етап можна перевірити окремо, що спрощує локалізацію помилок.

По завершенні керувальної дії система повертається до стану очікування HTTP-запиту, готова до наступного циклу взаємодії. Діаграму взаємодії процесів наведено на рисунку 3.3.

Діаграма наочно демонструє дворівневу логіку доступу: незахищений рівень (перегляд стану через /status) та захищений рівень (керувальні дії, що вимагають попередньої автентифікації). Така архітектура відповідає принципу найменших привілеїв і є типовою для IoT-систем з локальним REST API.

Процесна логіка системи побудована навколо повторюваного циклу запит-відповідь. Мобільний застосунок формує дію користувача у вигляді HTTP-запиту, сервер годинника приймає цей запит, перевіряє його, звертається до WatchState, формує JSON-відповідь і повертає результат клієнту. Після цього інтерфейс оновлює відображення відповідно до отриманих даних.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		43

В процесі автентифікації важливо зазначити, що перевірка PIN відбувається саме на стороні годинника. Мобільний застосунок лише збирає введений код і передає його серверу. Якщо PIN правильний, сервер встановлює `authenticated = true`. Якщо PIN не вірний, стан системи не змінюється. Така послідовність дозволяє уникнути ситуації, коли клієнт самовільно надає собі доступ.

Процес виконання захищених дій складається з кількох етапів: прийом POST-запиту, читання JSON-поля, перевірка формату даних, перевірка `authenticated`, оновлення відповідного поля стану та повернення відповіді. Така послідовність використовується для синхронізації часу та встановлення будильника. Вона є однаковою для симулятора і `firmware`-версії, що спрощує тестування.

Якщо користувач намагається виконати керувальну дію без автентифікації, сервер повертає помилку 403. Якщо введено неправильний формат часу, повертається 400. Якщо PIN неправильний, повертається 401. Використання різних HTTP-статусів дозволяє Android-застосунку точніше реагувати на помилки й показувати користувачу зрозуміле повідомлення.

Процес оновлення web UI відбувається паралельно з роботою мобільного застосунку. Браузерна сторінка періодично виконує функцію (GET `/status`) і відображає актуальний стан годинника. Тому зміни, виконані з застосунку, через короткий час стають видимими і у браузері. Це добре демонструє, що всі інтерфейси працюють з одним спільним станом пристрою.

Процес демонстрації системи можна описати як послідовність наступних дій: запуск `watch_sim`, відкриття web UI, запуск мобільного застосунку в емуляторі, введення PIN коду, отримання стану, синхронізація часу, встановлення будильника та блокування годинника. Такий сценарій охоплює всі основні функції системи й дозволяє показати її роботу за короткий час.

З точки зору тестування, кожний процес можна перевірити окремо. Спочатку перевіряється доступність `/status`, потім правильна та неправильна автентифікація, далі захищені операції `/sync-time` та `/alarm`, після чого перевіряється `/lock`. Такий підхід дозволяє швидко знайти місце помилки: у мобільному застосунку, у мережевому шарі або на стороні симулятора.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						44
Вим	Арк.	№ докум.	Підпис	Дата		

Обмеження поточного процесу полягають у тому, що PIN є статичним, трафік не шифрується, а сенсори симулюються програмно. Для дипломного прототипу це є допустимим, оскільки головна мета полягає у демонстрації архітектури, взаємодії компонентів і механізму автентифікації. У подальшому ці процеси можна розширити: додати реальні сенсори, захищене зберігання PIN, BLE-з'єднання або повноцінну роботу з фізичним дисплеєм.

К6ПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						45
Вим	Арк.	№ докум.	Підпис	Дата		

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Розробка блок-схем алгоритмів є важливим етапом документування програмного забезпечення, оскільки дозволяє наочно відобразити логіку функціонування системи та послідовність виконання операцій. У даному підрозділі описано два ключових алгоритми: алгоритм роботи основної програми (симулятора watch_sim / прошивки ESP32) та алгоритм автентифікації користувача через мобільний додаток.

Алгоритм роботи основної програми описує повний цикл функціонування серверної частини системи від запуску до завершення роботи. Програма реалізована у двох варіантах: локальний симулятор на C++ (watch_sim) та прошивка для мікроконтролера ESP32. Обидва варіанти реалізують ідентичну логіку та REST API, що забезпечує повну сумісність із мобільним додатком.

На початку роботи виконується ініціалізація структури даних WatchState, яка містить усі параметри стану годинника: час, дату, кількість кроків, рівень заряду батареї, температуру, час будильника та статус автентифікації. Після ініціалізації запускається HTTP-сервер на порту 8080, а також окремий потік для обробки клавіш керування. Ці дві підсистеми працюють паралельно в окремих потоках, не блокуючи основний цикл програми.

Основний цикл програми перевіряє прапорець g_running. Якщо значення true виконується оновлення симуляції сенсорів (кроки збільшуються кожні 5 секунд, рівень батареї зменшується кожні 20 секунд, температура змінюється кожні 15 секунд), після чого відбувається перемалювання термінального OLED UI та затримка 1 секунда перед наступною ітерацією. Якщо прапорець g_running встановлено у false – виконується коректне завершення всіх потоків та вихід з програми.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						46
Вим	Арк.	№ докум.	Підпис	Дата		

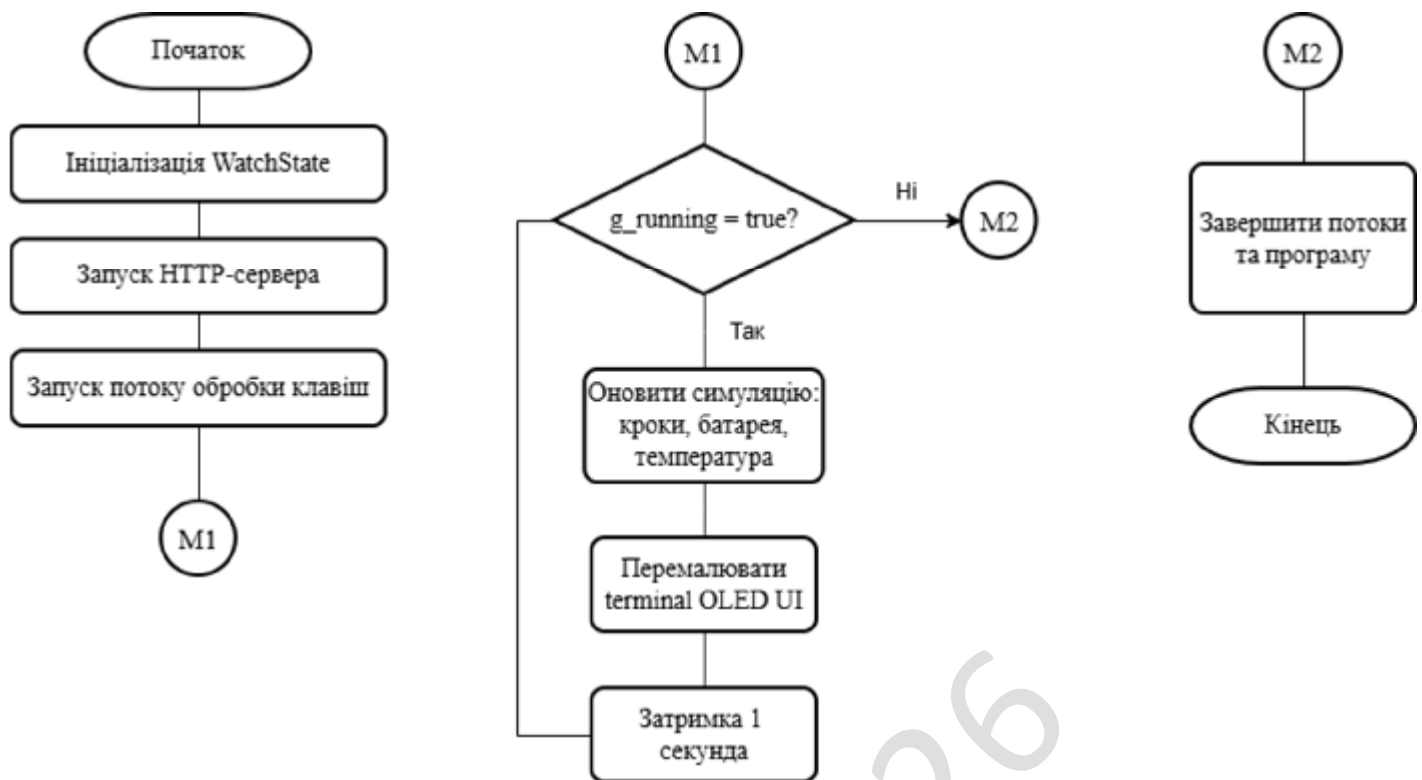


Рисунок 4.1 – Блок-схема алгоритму роботи основної програми

Як видно з блок-схеми (рисунок 4.1), алгоритм основної програми реалізує модель багатопотокового циклу з перевіркою умови продовження роботи. Ключовою особливістю є розподіл обов'язків між потоками: HTTP-сервер обробляє вхідні запити від мобільного додатку незалежно від основного циклу, який займається оновленням симульованих показників сенсорів та відображенням стану пристрою на екрані.

Така архітектура забезпечує стабільну роботу сервера та відповідає принципам побудови IoT-систем. Алгоритм автентифікації користувача описує послідовність дій при введенні PIN-коду в мобільному застосунку. Автентифікація є обов'язковим кроком перед отриманням доступу до керувальних функцій системи (синхронізація часу, встановлення будильника, блокування пристрою). Перегляд поточного стану годинника через ендпоінт /status доступний без автентифікації.

Алгоритм починається з введення користувачем чотиризначного PIN-коду в полі екрана LoginActivity мобільного додатку. Після натискання кнопки "Автентифікувати" застосунок перевіряє довжину введеного коду: якщо код не містить рівно 4 символи,

відображається повідомлення про помилку введення і процес повертається на початок.

Якщо перевірка пройдена, застосунок формує JSON-об'єкт {"pin": "XXXX"} і надсилає POST-запит до ендпоінту /auth локального HTTP-сервера.

Серверна частина (watch_sim або ESP32-прошивка) отримує запит та зчитує поле pin з тіла запиту. Отримане значення порівнюється зі збереженим демонстраційним PIN-кодом (1234).

У разі збігу сервер встановлює прапорець authenticated = true у структурі WatchState та повертає відповідь {"authenticated": true} зі статусом HTTP 200, мобільний застосунок відкриває екран DashboardActivity та надає доступ до всіх функцій керування. У разі невідповідності PIN-коду сервер встановлює authenticated = false та повертає відповідь {"authenticated": false, "error": "Invalid PIN"} зі статусом HTTP 401, а мобільний застосунок відображає повідомлення про невірний PIN-код.

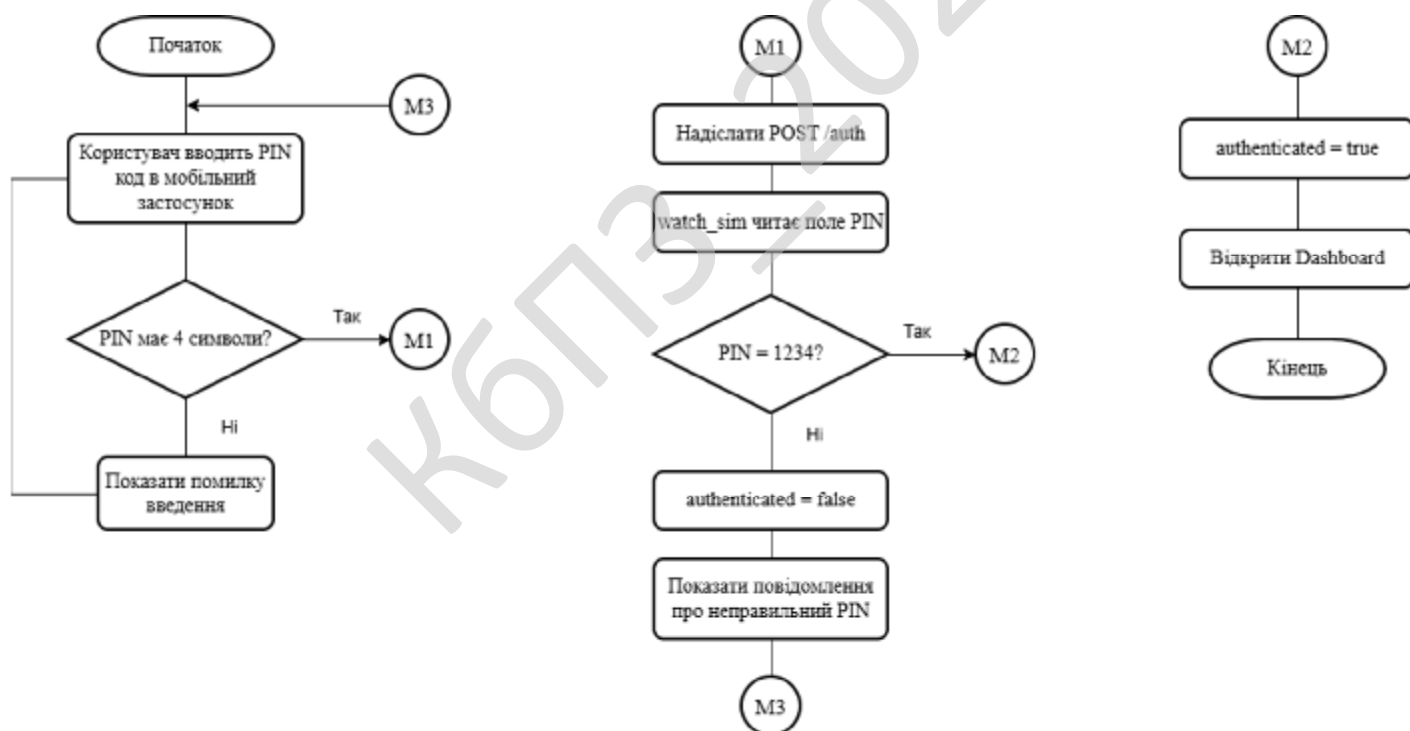


Рисунок 4.2 – Блок-схема алгоритму роботи автентифікації

Блок-схема алгоритму автентифікації (рисунок 4.2) наочно демонструє дворівневу перевірку: спочатку на стороні клієнта (довжина PIN), а потім на стороні

сервера (відповідність збереженому значенню). Така архітектура зменшує кількість зайвих мережесих запитів та забезпечує коректну обробку помилок введення ще до звернення до сервера. Підхід відповідає стандартним практикам побудови систем автентифікації для локальних IoT-пристроїв.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення реалізований на рівні автентифікації за допомогою PIN-коду. Система використовує механізм, при якому доступ до керувальних функцій (синхронізація часу, встановлення будильника, блокування пристрою) надається лише після успішної автентифікації. Без введення правильного PIN-коду користувач може лише переглядати поточний стан годинника через ендпоінт /status, але не може змінювати його параметри.

Серверна частина при отриманні POST-запиту до захищених ендпоінтів (/sync-time, /alarm) обов'язково перевіряє прапорець authenticated у структурі WatchState. У разі відсутності автентифікації сервер повертає HTTP-статус 403 Forbidden з повідомленням "Authenticate first", що унеможливорює несанкціоновану зміну параметрів пристрою. Архітектура системи відповідає принципу найменших привілеїв: незахищений рівень дозволяє лише читання стану, захищений – повне керування.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						49
Вим	Арк.	№ докум.	Підпис	Дата		

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Впровадження системи «розумний годинник» в промислову або дослідницьку експлуатацію передбачає послідовне виконання ряду організаційних і технічних заходів, спрямованих на забезпечення стабільної, безперервної та безпечної роботи розробленого програмного забезпечення.

Підготовка апаратного і програмного середовища

Перед розгортанням системи необхідно підготувати апаратну складову: зібрати електронний пристрій на базі мікроконтролера ESP32 з підключеними вбудованими модулями (сенсор температури/вологості), OLED-дисплеєм SSD1306 та акумулятором. Схема електричних з'єднань повинна відповідати проектній документації.

Прошивка мікроконтролера виконується через Arduino IDE (версія 2.x і вище) з встановленими бібліотеками: ESP32 Board Support Package, DHT sensor library (Adafruit), Adafruit GFX Library, Adafruit SSD1306, ArduinoJson. Завантаження прошивки здійснюється через USB-кабель за допомогою програматора CH340/CP2102 або через OTA (Over-The-Air) оновлення при повторних розгортаннях.

Мобільний застосунок Android встановлюється на смартфон шляхом встановлення APK-файлу або через публікацію у внутрішньому корпоративному магазині застосунків. Мінімальні вимоги до пристрою: Android 8.0 (API level 26) і вище, підтримка Wi-Fi 802.11 b/g/n, наявність не менше 50 МБ вільної пам'яті.

Конфігурування мережевої інфраструктури

Для коректної роботи системи ESP32 та смартфон з застосунком повинні знаходитись в одній локальній Wi-Fi мережі. У прошивці ESP32 налаштовуються SSID та пароль мережі. IP-адреса ESP32 може бути статичною або динамічно призначатися через DHCP. При використанні DHCP рекомендується закріпити за пристроєм фіксовану IP-адресу у налаштуваннях маршрутизатора (DHCP reservation) для стабільного з'єднання.

У застосунку BASE_URL налаштовується на IP-адресу ESP32 у локальній мережі. При необхідності організації віддаленого доступу поза межами локальної

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						50
Вим	Арк.	№ докум.	Підпис	Дата		

мережі можна використати VPN-тунель або технологію ngrok для проксування HTTP-трафіку. Для виробничого розгортання рекомендується перехід на HTTPS із самопідписаним або довіреним сертифікатом.

Тестування та верифікація системи

Перед введенням системи в експлуатацію проводиться комплексне тестування. Функціональне тестування включає перевірку всіх endpoint-ів HTTP API (GET /status, POST /auth, POST /sync-time, POST /alarm, POST /lock, POST /screen), перевірку коректності відображення даних на OLED-дисплеї та у мобільному застосунку, перевірку механізму автентифікації та блокування.

Навантажувальне тестування проводиться шляхом безперервного опитування сенсорів протягом 72 годин з метою виявлення можливих витоків пам'яті або нестабільностей у роботі мікроконтролера. Тестування відмовостійкості передбачає перевірку поведінки системи при відсутності Wi-Fi з'єднання: система має автоматично відновлювати підключення без втрати даних або аварійного завершення роботи.

Супровід та технічна підтримка

У рамках технічного супроводу системи передбачається: ведення журналу подій через Serial-інтерфейс з можливістю підключення до комп'ютера для моніторингу; документування всіх виявлених дефектів та їх усунення шляхом перепрошивки пристрою; проведення планового технічного обслуговування раз на квартал із перевіркою стану акумулятора, контактів та сенсорів.

Оновлення прошивки виконується через OTA-механізм (Over-The-Air) без необхідності фізичного підключення до комп'ютера. Оновлення мобільного застосунку виконується стандартними засобами Google Play або через ручне встановлення нового APK. Всі оновлення тестуються на стенді перед розгортанням на виробничих пристроях.

Результати тестування

Нижче представлено результати тестування роботи мобільного додатку та симулятора годинника, що підтверджують коректність реалізованих алгоритмів та функціональність системи в цілому.

На рисунку 5.1 зображений веб-інтерфейс локального симулятора системи розумний годинник, що відкривається у браузері за адресою <http://127.0.0.1:8080/>.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		51

Ліворуч відображається візуальна модель розумного годинника з поточними даними на "екрані" пристрою (час 16:01, 7536 кроків, Battery 30%, Temp 18 C, Alarm 06:45, статус UNLOCKED).

Праворуч розміщено панель з детальною інформацією про стан годинника та кнопками керування (Prev Screen, Next Screen, Lock/Unlock, Refresh Now), даний інтерфейс дозволяє демонструвати роботу системи без мобільного пристрою.

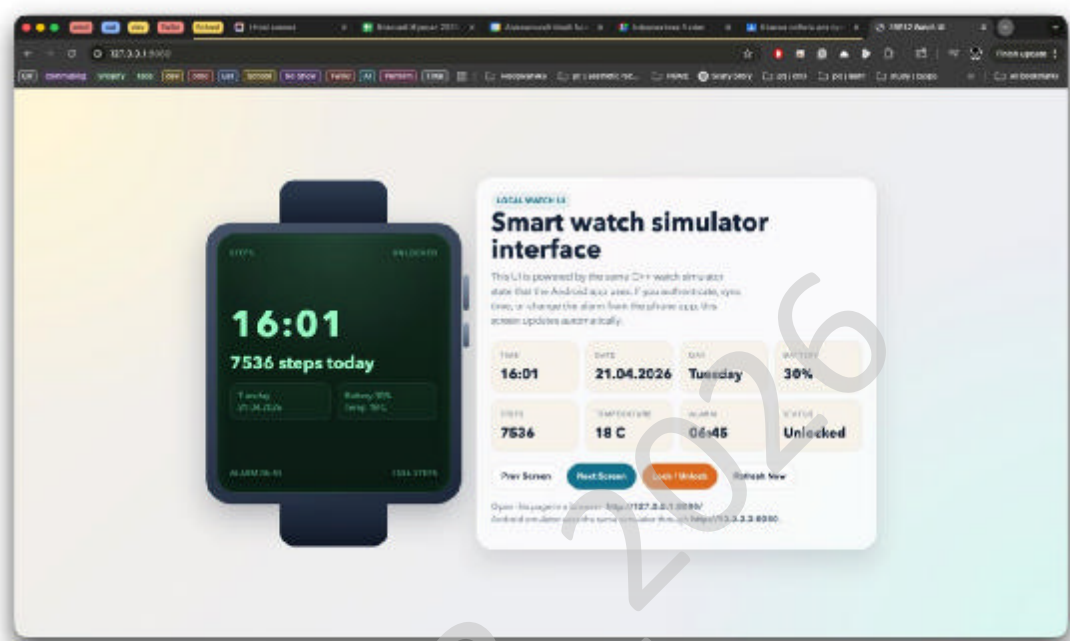


Рисунок 5.1 – Веб-інтерфейс симулятора розумного годинника

На рисунку 5.2 показана панель керування Watch Actions мобільного додатку після успішної автентифікації. Користувачу доступні наступні дії: отримання поточного стану годинника (FETCH WATCH STATUS), синхронізація часу (SYNC TIME), встановлення будильника (SET ALARM) та блокування пристрою (LOCK WATCH).

Кнопка блокування виділена червоним кольором для акцентування уваги користувача на те що вона відрізняється від інших, та завдяки червоному кольору сигналізує на те що вона блокує пристрій.

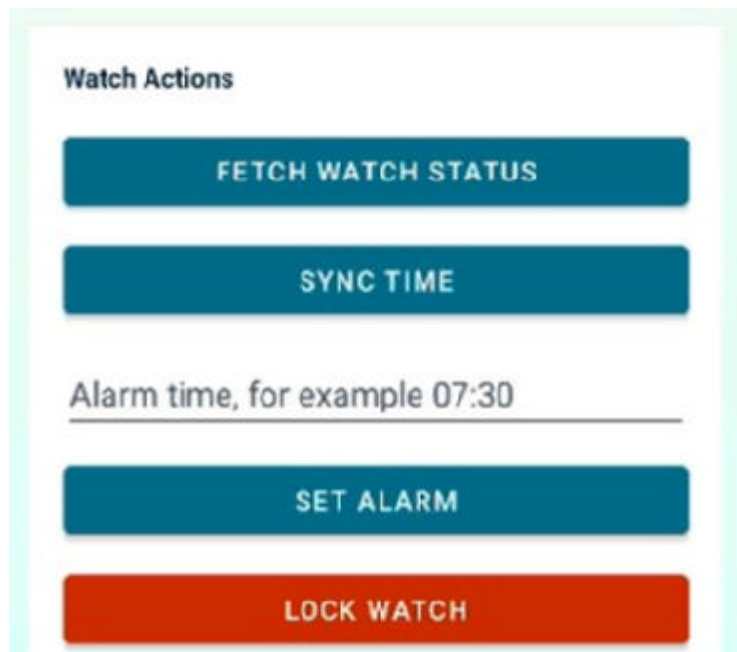


Рисунок 5.2 – Панель керування мобільного додатку

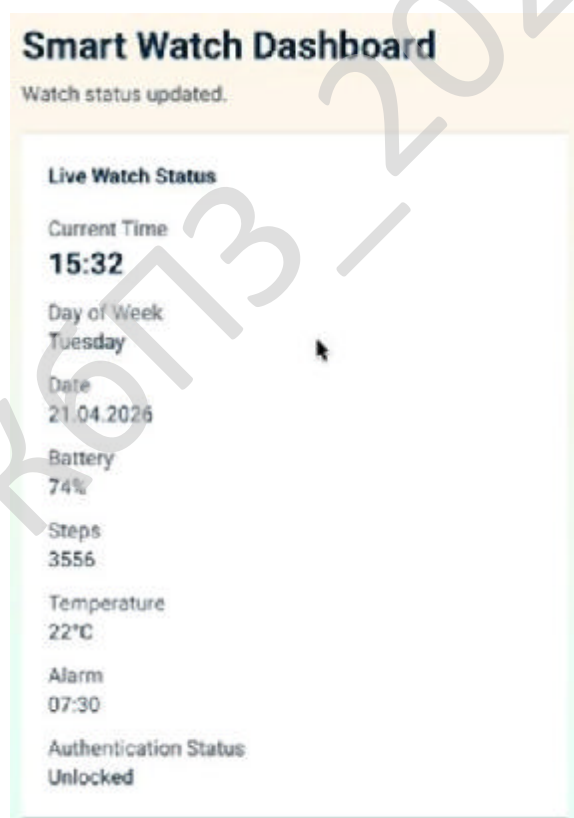


Рисунок 5.3 – Стан годинника у мобільному додатку

На рисунку 5.3 відображено розділ Live Watch Status панелі керування, що містить актуальні дані, отримані від сервера годинника через ендпоінт /status: поточний час (15:32), день тижня (Tuesday), дата (21.04.2026), рівень заряду батареї (74%), кількість кроків (3556), температура (22°C), час будильника (07:30) та статус автентифікації (Unlocked). Ці дані оновлюються при натисканні кнопки FETCH WATCH STATUS.



Рисунок 5.4 – Екран автентифікації мобільного додатку

На рисунку 5.4 зображено екран автентифікації мобільного додатку (LoginActivity). Користувач вводить чотиризначний PIN-код та натискає кнопку "АВТЕНТИФІКУВАТИ". Додаток підтримує перемикання мови інтерфейсу між українською та англійською мовами безпосередньо на екрані входу.

Результати тестування підтверджують коректну роботу всіх компонентів системи: мобільний застосунок успішно виконує автентифікацію та керування пристроєм, локальний симулятор коректно обробляє всі типи HTTP-запитів та підтримує актуальний стан годинника, а веб-інтерфейс забезпечує зручну демонстрацію функціональності системи без потреби у фізичному пристрої.

К6ПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						55
Вим	Арк.	№ докум.	Підпис	Дата		

6 ОСНОВНІ ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено програмне забезпечення інтелектуальної системи «розумний годинник» з автентифікацією даних через мобільний додаток. За результатами виконаної роботи можна зробити наступні висновки.

Виконано аналіз існуючих аналогів систем розумних годинників (Amazfit Bip 3 Pro, Samsung Galaxy Watch 7, Huawei Watch GT 5 Pro, Garmin Fenix 8), що дозволило визначити ключові функціональні вимоги до розробленої системи та обґрунтувати вибір технологічного стеку для її реалізації.

Обґрунтовано вибір апаратної платформи на базі мікроконтролера ESP32, який забезпечує вбудовану підтримку Wi-Fi та Bluetooth, достатню обчислювальну потужність для одночасної обробки сенсорних даних та HTTP-сервера, а також низьке енергоспоживання. Для вимірювання температури та вологості навколишнього середовища обрано сенсор DHT22, для виведення інформації – OLED-дисплей SSD1306.

Розроблено прошивку мікроконтролера мовою C++ в середовищі Arduino IDE. Прошивка реалізує: ініціалізацію периферійних пристроїв, підключення до Wi-Fi мережі з автоматичним відновленням з'єднання, HTTP API-сервер з шістьма endpoint-ами, зчитування та обробку даних з сенсора DHT22, симуляцію показників кроків та заряду батареї, виведення інформації на OLED-дисплей, а також PIN-код автентифікацію.

Розроблено мобільний застосунок мовою Kotlin для платформи Android, який забезпечує: автентифікацію користувача за PIN-кодом, відображення поточного стану годинника в реальному часі, синхронізацію системного часу з годинником, встановлення будильника та відправку push-сповіщень, блокування та розблокування пристрою, підтримку двох мов інтерфейсу (українська та англійська).

Розроблено та задокументовано ключові алгоритми роботи системи: алгоритм прошивки ESP32 та алгоритм мобільного застосунку, представлені у вигляді блок-схем відповідно до стандартів. Алгоритми забезпечують надійну та стабільну роботу системи

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		56

в умовах нестабільного мережевого з'єднання.

Реалізовано багаторівневу систему захисту від несанкціонованого доступу, яка включає PIN-автентифікацію, механізм блокування пристрою та обмеження комунікації в межах локальної мережі.

Практична цінність роботи полягає у демонстрації можливості побудови повноцінної IoT-системи на базі доступної апаратної платформи ESP32 з реалізацією клієнт-серверної архітектури без використання хмарних посередників. Розроблене програмне забезпечення може бути використане як основа для подальшого розвитку системи, зокрема додавання біометричних сенсорів, хмарного зберігання даних та машинного навчання для аналізу активності користувача.

КБПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						57
Вим	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Espressif Systems. ESP32 Technical Reference Manual. Version 5.7. Shanghai : Espressif Systems, 2026. 166-172 p.
2. Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry. 2nd ed. Indianapolis : Wiley, 2020. 366-383 p.
3. Horowitz P., Hill W. The Art of Electronics. 3rd ed. Cambridge : Cambridge University Press, 2015. 751-752 p.
4. Skeen J., Greenhalgh D. Kotlin Programming: The Big Nerd Ranch Guide. Indianapolis : Pearson Technology Group. 2018. 440-477 p.
5. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : PhD thesis. Irvine : University of California, 2000. 162 p.
6. Tanenbaum A. S., Wetherall D. Computer Networks. 5th ed. Upper Saddle River : Prentice Hall, 2011. 424-485 p.
7. Stroustrup B. The C++ Programming Language. 4th ed. Upper Saddle River : Addison-Wesley Professional, 2013. 1033-1049 p.
8. Meier R. Professional Android. 4th ed. Indianapolis : Wrox, 2018. 54-81 p.
9. Kurniawan A. Internet of Things Projects with ESP32. Birmingham : Packt Publishing, 2019. 179-190 p.
10. Lospinoso J. C++ Crash Course: A Fast-Paced Introduction. San Francisco: No Starch Press, 2019. 243-260 p.
11. Vermesan O., Friess P. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. Aalborg : River Publishers, 2013. 177-179 p.
12. Жураковський Б.Ю., Зенів І.О. ТЕХНОЛОГІЇ ІНТЕРНТУ РЕЧЕЙ : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 15-23 с.
13. Williams A. C++ Concurrency in Action Practical Multithreading. Shelter Island: Manning, 2012. 16-25 p.

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						58
Вим	Арк.	№ докум.	Підпис	Дата		

14. Donahoo M. J., Calvert K. L. TCP/IP Sockets in C: Practical Guide for Programmers. 2nd ed. Burlington : Morgan Kaufmann, 2009. 63 p.
15. Marrs T. JSON at Work: Practical Data Integration for the Web. Boston : O'Reilly Media, 2017. 85 p.
16. Minerva R., Biru A., Rotondi D. Towards a Definition of the Internet of Things (IoT). IEEE Internet Initiative. 2015. Vol. 1. 2–86 p.
17. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials. 2015. Vol. 17, № 4. 2347-2372 p.
18. Díaz M., Martín C., Rubio B. State-of-the-Art, Challenges, and Open Issues in the Integration of Internet of Things and Cloud Computing. Journal of Network and Computer Applications. 2016. 2-24 p.
19. Naik N. Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. IEEE International Symposium on Systems Engineering (ISSE). 2017. P. 1–7.
20. Ray P. P. A Survey of IoT Cloud Platforms. Future Computing and Informatics Journal. 2016. Vol. 1, № 1–2. P. 35–45.
21. Hasan M., Islam Md. M., Zarif M. I. I., Hashem M. M. A. Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches. Internet of Things. 2019. Vol. 7. Article 100059.
22. Atzori L., Iera A., Morabito G. The Internet of Things: A Survey. Computer Networks. 2010. Vol. 54, № 15. P. 2787–2805.
23. Tkachenko O., Palahin V. Authentication Methods for IoT Embedded Systems: Review. Journal of Information Technologies. 2021. Vol. 5, № 1. P. 44–52.
24. Espressif Systems. ESP-IDF Programming Guide – Stable Version. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html> (дата звернення: 09.04.2026).
25. Espressif Systems. ESP32 DEVKITV1 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 09.04.2026).

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		59

26. Espressif Systems. ESP-WROOM-32 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (дата звернення: 09.04.2026).
27. Arduino LLC. Arduino IDE 2.x Documentation. URL: <https://docs.arduino.cc/software/ide-v2> (дата звернення: 10.04.2026).
28. Arduino Libraries. ESP32 WebServer Library Reference. URL: <https://github.com/espressif/arduino-esp32/tree/master/libraries/WebServer> (дата звернення: 12.04.2026).
29. JetBrains. Kotlin Programming Language – Official Documentation. URL: <https://kotlinlang.org/docs/home.html> (дата звернення: 12.04.2026).
30. JetBrains. Coroutines Guide – Kotlin. URL: <https://kotlinlang.org/docs/coroutines-guide.html> (дата звернення: 13.04.2026).
31. Google LLC. Android Developer Guides – App architecture. URL: <https://developer.android.com/guide/components/activities/intro-activities> (дата звернення: 13.04.2026).
32. Google LLC. HttpURLConnection – Android API Reference. URL: <https://developer.android.com/reference/java/net/HttpURLConnection> (дата звернення: 13.04.2026).
33. Google LLC. NotificationCompat – Android Jetpack. URL: <https://developer.android.com/reference/androidx/core/app/NotificationCompat> (дата звернення: 16.05.2026).
34. Waveshare Electronics. 1.3inch OLED Module (SSH1106) Wiki. URL: https://www.waveshare.com/1.3inch-oled-b.htm?srsId=AfmBOopxogNinvXFwKSte_7GdJRV7FaPXRIfqmk-3w3r-5xHDz_MqhYP (дата звернення: 06.05.2026).
35. MP2606. Single Cell 1A Linear Li-Ion Battery Charger. URL: <https://www.monolithicpower.com/en/mp2606.html> (дата звернення: 07.04.2026).
36. Amazfit. Amazfit Bip 3 Pro. URL: https://www.amazfit.com/products/amazfit-bip-3-pro#/ (дата звернення: 20.04.2026).

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		60

37. Apple Inc. Посібник користувача Apple Watch. URL: <https://support.apple.com/uk-ua/guide/watch/welcome/11.0/watchos/11.0> (дата звернення: 21.04.2026).

38. Samsung Electronics. Смарт-годинник Samsung Galaxy Watch7. URL: <https://www.samsung.com/ua/watches/galaxy-watch/galaxy-watch7-40mm-cream-lte-sm-1305fzeasek/> (дата звернення: 20.04.2026).

39. Huawei Technologies. Huawei Watch GT 5 Pro. URL: <https://consumer.huawei.com/ua/wearables/watch-gt5-pro/> (дата звернення: 21.04.2026).

40. Garmin Ltd. Garmin Fenix 8 – Product Specifications. URL: <https://support.garmin.com/en-US/?partNumber=010-03199-00&tab=manuals> (дата звернення: 22.04.2026).

41. Oracle Corporation. Java SE 8 – Class Thread Documentation. URL: <https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html> (дата звернення: 26.04.2026).

42. Microsoft. Visual Studio Code – Code Editor. URL: <https://code.visualstudio.com/docs> (дата звернення: 10.04.2026).

43. Google LLC. Android Studio – Official IDE for Android Development. URL: <https://developer.android.com/studio/intro> (дата звернення: 10.04.2026).

44. Vermesan O., Friess P. Internet of Things – From Research and Innovation to Market Deployment. Aalborg : River Publishers, 2014. 355 p. URL: https://www.riverpublishers.com/pdf/ebook/RP_E9788793102958.pdf (дата звернення: 17.04.2026).

45. Wikipedia. Розумний годинник. URL: https://uk.wikipedia.org/wiki/Розумний_годинник (дата звернення: 15.04.2026).

46. Wikipedia. Годинник. URL: <https://uk.wikipedia.org/wiki/Годинник> (дата звернення: 15.04.2026).

47. GitHub Inc. espressif/arduino-esp32 – Arduino Core for the ESP32. URL: <https://github.com/espressif/arduino-esp32> (дата звернення: 10.04.2026).

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
Вим	Арк.	№ докум.	Підпис	Дата		61

48. MDN Web Docs. HTTP response status codes. Mozilla Foundation. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status> (дата звернення: 08.04.2026).

49. Android Open Source Project. Android Security. URL: <https://source.android.com/docs/security> (дата звернення 19.04.2026)

50. IT community. ТОП 10 розумних годинників на всі випадки життя. URL: <https://itc.ua/ua/articles/top-10-rozumnyh-godynnykiv-na-vsi-vypadky-zhyttya/> (дата звернення: 19.04.2026).

51. Wikipedia. ESP32. URL: <https://en.wikipedia.org/wiki/ESP32> (дата звернення: 19.05.2026).

52. Google. Pixel Watch 4. URL: https://store.google.com/us/product/pixel_watch_4?hl=en-US (дата звернення 19.04.2026)

К6ПЗ_2026

					ВКРБ-122.26.0016.00.00.ПЗ	Арк.
						62
Вим	Арк.	№ докум.	Підпис	Дата		