

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

До захисту допущено:

Завідувач кафедри

Сергій Стіренко

«___» _____ 2021 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: «Система тихого розумного будинку»

Виконав:

студент IV курсу, групи ІО-71

Єрмоленко Віталій Русланович _____

Керівник:

Ст. викладач

Аленін Олег Ігорович _____

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н.,

Сімоненко Валерій Павлович _____

Рецензент: _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2021 рік

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
Сергій СТИРЕНКО

(підпис)

“__” _____ 2021 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Єрмоленка Віталія Руслановича

1. Тема проєкту Система тихого розумного будинку, керівник проєкту Аленін Олег Ігорович, затверджені наказом по університету від _____ 2021 року № _____
2. Термін здачі студентом закінченого проєкту _____ 2021 р.
3. Вихідні дані до проєкту див. технічне завдання
4. Зміст пояснювальної записки (перелік питань, які розробляються)
Опис та аналіз предметної області, дослідження аналогів у даній області. Дослідження можливих варіантів апаратної реалізації та підбір релевантної апаратної частини. Розробка програмного забезпечення та програмно-апаратного комплексу
5. Перелік графічного матеріалу (з точним позначенням обов’язкових креслень) функціональна схема, принципова схема, структурна схема

6. Консультанта проекту, з вказівкою розділів проекту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ П/П	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2020-15.12.2020</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2020-15.03.2021</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2021-25.03.2021</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2021-5.04.2021</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2021-15.04.2021</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2021-20.05.2021</i>	
7.	<i>Захист програмного продукту</i>	<i>25.04.2021</i>	
8.	<i>Передзахист</i>	<i>23.05.2021</i>	
9.	<i>Захист</i>	<i>14.06.2021</i>	

Студент-дипломник _____
(підпис)

Керівник проекту _____
(підпис)

Анотація

Головна ідея роботи — розробка системи “тихого розумного будинку”, що була б здатна зайняти порожню нішу інклюзивності серед систем домашньої автоматизації. Головною метою проєкту є створення концепції інклюзивної системи без голосового керування, а також розробка робочого прототипу програмно-апаратного комплексу даної системи, який був би простим та інтуїтивно-зрозумілим у користуванні і мав можливості масштабованості системи за рахунок розширення асортименту модулів, як апаратних, так і програмних.

У даній роботі розглядаються основні тенденції як сучасного типу побудови подібних систем, так і загальноісторичні, для того, аби розроблена система мала велику релевантність із потребами за вимогами сучасного користувача.

В результаті роботи було створено прототип системи, що відповідає початковим завданням та створює значні можливості для подальшого розширення та добудови системи.

Annotation

The main idea of the study was the development of the “Silent Home System” that could take an empty inclusive niche among the Smart Home automatization systems. The main goal of the project is to develop the conception of inclusive non voice operated system, as well as to create the functioning prototype, both software- and hardware- wise, that would be simple and easy to use and could provide an opportunity to scale the system in terms of increasing the range of modules’ stock, both software- and hardware-based.

This study inspects the main tendencies of the modern and generalized development of the similar systems to make sure the system created is highly relevant to needs and demands of modern users.

As a result of the study, a working prototype has been developed that meets the initial requirements, as well as creates significant opportunities for further development and improvement of the system.

**Опис альбому
до дипломного проекту
на тему: «Система тихого розумного будинку»**

Київ – 2021 рік

№ з/п	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість	№	Примітки
1	A4	ІАЛЦ.467200.001 ВП	Система тихого Розумного будинку	1		
			Відомість проекту			
2	A4	ІАЛЦ.467200.002 ТЗ	Система тихого Розумного будинку	3		
			Технічне Завдання			
3	A4	ІАЛЦ.467200.003 ПЗ	Система тихого Розумного будинку	77		
			Пояснювальна записка			
4	A4	ІАЛЦ.467200.004 Д1	Система тихого Розумного будинку	1		
			Принципова схема			
5	A4	ІАЛЦ.467200.005 Д2	Система тихого Розумного будинку	1		
			Функціональна схема			
6	A4	ІАЛЦ.467200.006 Д3	Система тихого Розумного будинку	1		
			Структурна схема			

					ІАЛЦ.467200.001 ВП			
Змн.	Арк.	№ документа	Підпис	Дата				
Розроб.		Єрмоленко В.Р.			Система тихого розумного будинку Відомість проекту	Літ.	Аркуш	Аркушів
Перев.		Аленін О. І.					1	1
Реценз.						КПІ імені Ігоря Сікорського, ФІОТ, Група ІВ - 71		
Н. Контр.		Сімоненко В.П.						
Затверд.		Стіренко С.Г.						

**Технічне завдання
до дипломного проєкту**
на тему: «Система тихого розумного будинку»

Київ – 2021 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	2
2. ПРИЧИНИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до розробленого серверу.....	2
5.2. Вимоги до програмного забезпечення.....	3
5.3. Вимоги до апаратної частини.....	3
6. ЕТАПИ РОЗРОБКИ.....	3

					ІАЛЦ.467200.002 ТЗ						
Змн.	Арк.	№ документа	Підпис	Дата	Система тихого розумного будинку			Літ.	Аркуш	Аркушів	
Розроб.		Єрмоленко В. Р.								1	3
Перев.		Аленін О. І.									
Реценз.											
Н. Контр.		Сімоненко В.П.								КПІ імені Ігоря Сікорського, ФІОТ, Група ІО - 71	
Затверд.		Стіренко С.Г.									

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Це технічне завдання поширюється на розробку системи тихого розумного будинку. Область застосування: автоматизація рутинного домашнього людського побуду.

2. ПРИЧИНИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проєкту по освітньо-професійної програми “Комп’ютерні системи та мережі” спеціальності 123 “Комп’ютерна інженерія”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”.

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проєкту є розробка системи тихого розумного будинку, що буде автоматизувати певні домашні процеси без голосових команд.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література по комп’ютерних мережах, інтернету речей та домашній автоматизації, публікації в періодичних виданнях, довідники, публікації в Інтернеті з цих питань тощо.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленої системи

- Простота у використанні
- Можливість доповнення функціоналу за рахунок нових модулів або доповнення існуючих.
- Коректна обробка сигналів

					ІАЛЦ.467200.002 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

- Стабільність роботи системи

5.2. Вимоги до програмного забезпечення

- Операційна система на основі GNU/Linux
- Інтерпретатор мови Python 3
- Інтерпретатор мови MicroPython
- Бібліотека MicroPython для роботи з ридером MFRC522

5.3. Вимоги до апаратної частини

- Мікрокомп'ютер Raspberry Pi Zero W
- Вільний простір твердотільного накопичувача не менше 1 Гбайт
- Мікроконтролер на базі ESP8266 із більш ніж 12 доступними пінами
- RFID ридер RC522

6. ЕТАПИ РОЗРОБКИ

Дата

6.1 Вивчення літератури

6.2 Складання і узгодження технічного завдання

6.3 Написання вступної частини та огляд рішень

6.4 Виокремлення власної концепції та підбір апаратної частини

6.5 Написання програмної частини

6.6 Тестування і виправлення помилок

6.7 Оформлення документації дипломного проекту

					ІАЛЦ.467200.002 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

**Пояснювальна записка
до дипломного проекту**
на тему: «Система тихого розумного будинку»

Київ – 2021 рік

ЗМІСТ

ВСТУП.....	2
РОЗДІЛ 1 АНАЛІЗ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ.....	3
1.1 Історична генеза "Розумного будинку".....	3
1.2 Сучасний "Розумний будинок".....	14
1.3 Огляд існуючих систем.....	15
Висновок до розділу 1.....	18
РОЗДІЛ 2 ПІДБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЗАДАЧІ.....	19
2.1 Особливості розробки ідеї.....	19
2.2 Вибір апаратної платформи хабу.....	20
2.2.1 Платформа Arduino.....	20
2.2.2 Серія мікроконтролерів ESP.....	27
2.2.3 Мікрокомп'ютери Raspberry Pi.....	32
2.3 Вибір RFID ридера.....	42
2.4 Вибір мікроконтролера для ридера.....	44
Висновок до розділу 2.....	47
РОЗДІЛ 3 ОПИС РЕАЛІЗАЦІЇ СИСТЕМИ ТИХОГО РОЗУМНОГО БУДИНКУ.....	48
3.1 Дверний модуль.....	48
3.2 Хаб.....	62
3.3 Модулі тестування.....	69
Висновок до розділу 3.....	71
ВИСНОВОК.....	72
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	74

					ІАЛЦ.467200.003 ПЗ						
Змн.	Арк.	№ документа	Підпис	Дата	Система тихого розумного будинку			Літ.	Аркуш	Аркушів	
Розроб.		Єрмоленко В. Р.								1	77
Перев.		Аленін О. І.									
Реценз.											
Н. Контр.		Сімоненко В.П.								КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО - 71	
Затверд.		Стіренко С.Г.									

ВСТУП

У сучасному світі діджиталізація із кожним днем все сильніше і сильніше охоплює всі прошарки людського існування. Чи не щопівроку проходять найрізноманітніші конференції, на яких програмісти, інженери, маркетологи представляють широкому загалу свої найвідвертіші та найнеймовірніші концепції, що мають спростити життя пересічному гомосапієнсу у повсякденні, завдяки новим технологіями. Із кожним місяцем нові технології все глибше і глибше інтегруються у наше повсякденне життя. Іще у якомусь 2009 році людина із кишеньковим персональним комп'ютером мала статус, щонайменше, "гіка", а у 2021 році нам буде вже досить важко знайти того, хто б не мав смартфона.

Те, що ще якихось років 10 тому здавалося маркером достатку та успішності, наразі не викликає якогось здивування і, радше, навпаки, відсутність цієї глибокої інтеграції інформаційних технологій викликає у пересічних людей щире зачудування причинами цієї прикрості.

Тож зовсім не дивним наразі є той факт, що і звичайнісінькі, здавалося б, домашні речі, починають швидко ставати "розумними", підключатися до інтернету та робити усе можливе для автоматизації нашого із вами життя.

Метою даної роботи є дослідження концепції "розумного будинку" та її найбільш успішних представників на світовому та Українському ринках, пошук порожніх ніш, у яких не представлені системи зі специфічними задачами та нового підходу до побудови подібних систем із наступною програмно-апаратною реалізацією проекту такої системи.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

РОЗДІЛ 1 АНАЛІЗ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Історична генеза "Розумного будинку"

Безперечно, надшвидка технологічна експансія, що призвела до глибокої інтеграції різноманітних комп'ютерно-обчислювальних систем у повсякденне життя людини, відбулася завдяки багатьом факторам, проте найбільше, напевно, варто завдячувати надзвичайно швидкому розвитку і поширенню мережевих технологій. Як можна судити зі статистики Міжнародного союзу електрозв'язку при ООН [14], темпи розвитку інтернет-сполучення, а також неймовірний ріст кількості пристроїв, що мають до нього доступ, є рушійною силою цього прогресу, а також очевидно показують усе більшу доступність як Інтернету загалом, так і складних комунікаційно-обчислювальних пристроїв зокрема. Тож зовсім не дивним є те, що інженерам на думку прийшло підключати до мережі ще й інші пристрої для автоматизації великої кількості процесів та отримання максимальної вигоди від обробки різних об'ємів інформації.

Колосальні темпи технологічного розвитку, а також прийняття технологій широким загалом протягом останніх трьох декад, що цей розвиток тільки підбурювали, з кожним роком рухати технології усе далі і далі, і через їх комерціалізацію і значний попит на ринку привели нас із Вами у точку історії, коли навіть електрочайник може бути "розумним", мати підключення до інтернету, дистанційне керування і хто зна що ще. На зорі "інформаційної ери" науковці і замислитись навряд могли про те, що обчислювальна техніка буде поміщатися у людську долоню, і мати потужність у 1250 гігафлопс.

Ріст розвитку, виробничих потужностей, комерціалізації і попиту призвели до того, що із кожним роком комплектуючі для обчислювальної техніки стають усе більш дешевими, при чому аніскільки не втрачаючи свій потенціал, а часто навіть навпаки - нарощуючи його, а необхідність у

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

аварійного скидання тиску для забезпечення цілісності та безпеки системи. Звісно ж, цей “мозок”, за сучасними оцінками є цілком примітивним блоком управління, що складається із друкованої плати і пригоршні радіоелементів, проте не можна недооцінювати роль цього нововведення у еволюції того, що нині ми іменуємо системою “Розумного дому”. Адже ці, простенькі на перший погляд, “мізки” на довгі роки задали вектор розвитку систем, що мають допомагати людині у побуті.

Першою ж краплею для сучасного океану даних систем варто вважати X10. X10 - це міжнародний відкритий протокол для комунікації між електронними девайсами домашньої автоматизації [16]. Він був розроблений в 1975 році і використовував для передачі сигналів електропроводку і бездротові канали (тобто радіосигнал). Керуючі сигнали між модулями системи у цьому протоколі передаються за допомогою короткочасних спалахів радіочастотних імпульсів, частотою у 120 кГц і тривалістю в 1 мс, або через електромережу таким самим чином. Не залежно від середовища передачі, команди X10 складаються наступним чином:

- 4 біти - код будинку
- 4 біти - код модуля, або групи модулів
- 4 біти - команда

Кожен модуль реагує тільки на команди, направлені відповідно до нього, або за кількома ширококомовними командами. Наприклад, мережею може прийти повідомлення “Модуль P10” (де P - код будинку, 10 - код модуля) із командою “увімкнути”, що змусить модуль P10 увімкнути приєднаний до нього пристрій.

Звісно, даний протокол надзвичайно простий по своїй суті, був достатньо передовим у свій час, хоча і не був ідеальним. Так, робота протоколу доволі повільна - три чверті секунди займає передача адреси

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

пристрою та команди, що може й не надто помітно при використанні настільного контролера для керування лише одним модулем, проте досить відчутно у взаємодії із інтелектуальним контроллером, що виконує певні сценарії для багатьох модулів. Система не може адекватно працювати з багатьма командами одночасно через колізії, викликані способом передачі команд. До того ж, різноманітні пристрої електромережі (наприклад, захист від перепадів напруги чи аварійного вимкнення живлення) можуть послаблювати сигнал настільки, що модулі не зможуть його зчитати, а також модулі погано працюють із пристроями низького споживання енергії.

Тим не менш, завдячуючи своїй простоті, а також масштабованості системи (адже вона є модульною і всі блоки керування знаходяться між точкою живлення та, власне, приладом, а отже можна у майбутньому докупити модулів для інших приладів і розширити домашню систему) модулі, що використовують протокол X10 є популярними і зараз, хоча на ринку є набагато більш розвинені варіанти із ширшими можливостями, проте X10 досить добре підходить для реалізації простенької системи “розумного будинку”, якщо запити користувача обмежуються дистанційним (у межах будинку) керуванням різними приладами та, можливо, досить простими сценаріями.

Тож ще у 1975 році був закладений напрямок до децентралізації системи, масштабованості, а отже і модульності та бездротовості, що і нині прослідковується у сучасних системах, про які мова піде нижче.

Наступним суттєвим етапом у побудові концепції “розумного будинку” такого вигляду, у якому вона існує зараз, стало широке розповсюдження доступу до мережі Інтернет серед пересічного населення.

Раніше всі елементи “домотики” були ніби “законсервовані” в межах однієї оселі, будинку чи комплексу. Так, вони вже використовували бездротові технології передачі сигналів, керувалися радіо-хвилями, і,

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

теоретично, керування було можливо і із-за меж будівлі, проте, така можливість була дуже обмежена через відстань розповсюдження керуючих сигналів. Тобто, скажімо, відчинити автоматичні ворота гаража зі свого автомобіля, що стоїть поруч із ними, було досить можливо, і, більш того, навіть реалізовано, проте, що робити, якщо тобі треба зробити цю ж дію, наприклад, із іншого континенту?

Саме цю проблему вирішував доступ до інтернету як користувача, так і самої системи. Тепер можна було керувати нею із будь-якої точки світу, якщо, звісно ж, там є підключення до всесвітньої мережі, і не боятися, що ваш бойлер у Г'юстоні залишився увімкненим, а ви знаходитесь у відрядженні у Брістолі і не можете цей факт ані перевірити, ані виправити.

Ця ідея долучила “розумний будинок” до концепції “інтернету речей”.

До того ж, розширення місць, придатних до керування системою, не стало єдиним нововведенням. Адже пов'язати мережею можна не тільки користувача та систему, але і окремі модулі системи між собою, для пришвидшення передачі даних, а також убезпечення безпеки. Звісно, нікому не потрібна інформація із вашого термостата, що висить за вікном, проте не досить приємно прокидатися посеред ночі від того, що якийсь жартівник вмикає і вимикає світло у вашій квартирі, просто підібравши необхідну команду радіохвиль (наприклад, якщо ви користуєтеся протоколом X10, що був описаний вище, то зловмиснику достатньо знати номер вашого будинку, аби керувати модулями системи). Звісно, не можна напевно сказати, що впровадження “інтернету речей” значно запобігає таким діям зловмисника, недарма ж світове співтовариство повсякчас гуде про приватність та вразливість даних у таких системах, проте - однозначно робить таке втручання набагато складнішим.

Загалом, девайси “інтернету речей” тут використовуються для контролю та моніторингу різноманітних механічних та електронних систем.

										Арк.
										8
Змн.	Арк.	№ докум.	Підпис	Дата						

У такому контексті виділяють три основні напрямки [17], або ж сфери використання:

- Інтеграція у систему енергоменеджменту будинку із метою створення ІОТ-керованого енергоефективного “розумного дому”.
- Можливі засоби моніторингу в режимі реального часу для зменшення споживання енергії та контролю поведінки мешканців.
- Інтеграція розумних пристроїв у вбудоване середовище та можливе їх використання у додатках.

Потенційно такі девайси можуть надати користувачу значні переваги за рахунок можливості їх зв'язку із іншими пристроями інфраструктури “інтернету речей”, зокрема з ІоМТ (Internet of Medical Things) для контролю стану здоров'я користувача та своєчасного надання йому необхідних медичних заходів та послуг, із транспортними та V2X системами, а також багатьма іншими. У далекій перспективі таке спілкування систем та девайсів “розумного дому” із іншими системами “інтернету речей” та їх суцільна взаємопов'язаність і обмін даними надають користувачам купу різноманітних можливостей у різноманітних сферах людського життя:

- Медичне обслуговування: лікарі зможуть використовувати зібрані системами дані про людську поведінку задля збору більш точного анамнезу і аналізувати малопомітні для самого суб'єкта речі, як, скажімо, підвищена сонливість та загальна слабкість при депресії, про що розкажуть дані ранішніх підйомів та іншої активності (якщо, звісно ж, будуть належним чином оброблені відповідним програмним забезпеченням).
- Сфера послуг: наприклад, система зможе оперативно повідомити користувача про прибуття кур'єра за відповідний час до того, як він подзвонить у двері, для того, аби для користувача це не стало

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- несподіванкою, якщо він чимось заклопотаний у даний момент; або ж, за неможливості відірватися від нагальних справ, кур'єра буде повідомлено про це і запропоновано змінити графік доставки для того, аби замовлення потрапило до клієнта у більш зручний час. До того ж інформація про користувача може допомогти підібрати більш релевантну рекламу, базуючись, скажімо, на даних про споживання кави, а також його зменшення після придбання певного сорту. В такому випадку реклама більше не пропонуватиме користувачу цей сорт бобів, і до того ж, буде показуватися саме тоді, коли запаси кави користувача будуть на грані закінчення.
- Соціальна сфера: “розумний дім” підключений до загальноміської системи [7] “інтернету речей” зможе оперативно та зручно інформувати користувача про трафік у місті, інциденти на дорогах, ремонтні роботи, значно полегшувати розрахунок часу виїзду та маршруту до пункту призначення, а також заздалегідь зачиняти вікна при підвищенні рівня забрудненості повітря та вмикати систему його очищення у домі.
- Безпека: система “розумного будинку” у разі несанкціонованого проникнення третіх осіб до приміщення зможе оперативно повідомити про це місцевий відділок поліції, надавши їм необхідні дані про місце, особи порушників, а також, за необхідності, затримати їх усередині до прибуття наряду.
- Догляд за хворими [12] та старими людьми: користування такими системами може значно спростити і без того складне життя інвалідів та людей літнього віку в досить очевидні способи, а також може надавати відповідним службам та близьким людям користувача необхідні дані про настрій, здоров'я та

					ІАЛЦ.467200.003 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

- життєдіяльність загалом, дозволяючи більш ефективно піклуватися про них. Голосовий контроль може допомогти користувачам з обмеженнями зору та рухливості, тоді як системи оповіщення можуть бути підключені безпосередньо до кохлеарних імплантів, які носять користувачі з вадами слуху [6]. Вони також можуть бути оснащені додатковими функціями безпеки. Ці функції можуть включати датчики, які контролюють наявність надзвичайних ситуацій, таких як падіння або судоми. [20] Застосована таким чином технологія розумного будинку може надати користувачам більше свободи та вищу якість життя.
- Енергетичний менеджмент: як було вже описано вище, система може значно допомогти у зниженні енергетичних витрат, оптимізувавши споживання енергії різного роду, до того ж, дані, отримані виробником енергії від кінцевих споживачів можуть допомогти йому оптимізувати системи розподілу та виробництва, а тако краще зрозуміти потреби.
- Моніторинг навколишнього середовища: отримуючи інформацію від систем моніторингу, “розумний будинок” може повідомити як про маленькі, проте досить значні речі, як то зниження якості питної води (і організувати доставку питної води до користувача, чи її фільтрації), так і попередити “великі” негаразди, наприклад торнадо чи землетруси, чим, певно, збереже не тільки самопочуття і майно, а і життя користувача.

Усе це може бути можливим не просто через, так би мовити, “спілкування” між собою різноманітних систем, тобто обмін сухими даними, а й через їх правильну обробку, що наразі усе більше віддається не складним алгоритмам, а штучному інтелекту та нейронним мережам, які, в оглядному майбутньому, можуть більш точно та ефективно обробляти надану їм

					ІАЛЦ.467200.003 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

інформацію і робити достатньо точні передбачення, а також давати поради по різних сферах.

Тим не менш, не обходиться і без шквальної критики цих технологій та концепцій. На думку багатьох людей усі ці надзвичайно передові речі, що мають вже в найближчому майбутньому значно покращити людське існування, є надто “молодими”, тобто ще дуже свіжі для того, аби укріпитися у свідомості пересічної людини як “надійні”. А відомі всім байки про “злих гакерів” (що ви могли неодноразово бачити у кінематографі), а також відомі вже далеко не всім, а лише людям, що у темі розбираються ґрунтовно (а зазвичай ще й працюють у цій сфері) випадки зламу пристроїв “інтернету речей” із метою майну криптовалюти чи проведення DDoS-атак, що мали місце досить нещодавно, значною мірою знижують популярність цих технологічних ідей у суспільстві.

На даний момент IoT страждає від фрагментації платформи, відсутності загальних технічних стандартів та сумісності. За словами Майкла Уолеса (Майкл Уоллес в даний час є старшим віце-президентом і генеральним менеджером Qualcomm в галузі обчислювальної техніки та XR бізнесу): “Фрагментація є ворогом Інтернету речей”[21].

Проте повернемося до “розумного будинку” та його побудови. Зазвичай система “Розумного будинку” є децентралізованою і складається не з одного модуля, що виконує всі функції, а із багатьох, в основному - вузькоспеціалізованих, модулів, кожен із яких виконує свою певну функцію. Як правило, аби не ускладнювати та кратно не здорожчувати систему, модулі, при їх спеціалізації, самостійними не є, і не можуть працювати без центрального пристрою, що іменується “хабом” або “гейтвеєм”.

Основне завдання “хабу” є обробка інформації, зібраної іншими модулями системи. До того ж, на нього можуть бути покладені і інші завдання, окрім обробки даних із інших пристроїв. “Хаб” може забезпечувати

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

з'єднання системи із мережею інтернет, виконувати функції вай-фай роутера для інших пристроїв, що не обов'язково є частинами системи, мати інтерфейс для внесення корективів у роботу систему та виведення інформації (наприклад - банальний тачскрін із інформацією про систему та налаштуваннями) і таке інше. Він зазвичай виконується у форматі настінного терміналу, або ж окремого модуля, що може як мати, так і не мати фізичного інтерфейсу користувача. В другому випадку виробники даних систем зазвичай пропонують користувачу такі варіанти як: доступ до інтерфейсу користувача із ПК або інших обчислювальних пристроїв, мобільний додаток, чи веб-інтерфейс.

Очевидною деталлю безперечно є те, що наявності самих лише пристроїв, або ж модулів розглядаємої системи не є достатньою умовою її функціонування, оскільки величезну роль у цьому відіграє програмне забезпечення системи, а також зв'язок усіх її компонентів між собою з метою обміну даними. Отже, можна з упевненістю зазначити, що основними складниками системи “Розумного будинку” є такі компоненти:

- апаратна частина, що складається із різноманітних модулів та “хабу”, який оброблює надану йому інформацію;
- програмне забезпечення, відповідальне за функціонування системи та обробку даних, і керування модулями;
- та протоколи зв'язку, по яких ведеться обмін даними та керуючими сигналами між компонентами системи.

До того ж, при класифікації пристроїв “розумного будинку” час від часу виділяють окремі модулі системи у специфічний клас, що називають “Актuatorами”.

Актuatorи - це усі виконавчі пристрої, які безпосередньо виконують команди системи. При такій класифікації дана група модулів стає найбільш численною, оскільки туди входять автоматичні вимикачі, автоматичні

					ІАЛЦ.467200.003 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

розетки, клапани труб, сирени, клімат-контролери і так далі.

1.2 Сучасний "Розумний будинок"

Отже, на момент 2021 року, "розумний будинок" зазвичай виділяють як підмножину "інтернету речей", оскільки найпоширенішим є доступ до пристроїв системи через мережу Інтернет. Загалом же, модулі у межах будинку здійснюють обмін інформацією між собою застосовуючи бездротові технології, тобто за допомогою радіосигналів, використовуючи такі стандарти:

- Z-Wave
- ZigBee
- Wi-Fi
- Bluetooth

А вже для зв'язку із зовнішнім світом контроллер підключається до мережі Інтернет.

Система "розумного будинку" має автоматизувати більшу частину людського рутинного побуту, а також гнучко підлаштовуватися під задачі кожного окремого користувача, якщо вони є чимось специфічними.

Також особливу увагу наразі загострено на загальній сек'юрності системи, як "зовнішній", так і "внутрішній". Тобто система має не тільки бути здатною захищати користувача та його фізичне майно, але й дані, що циркулюють системою, до яких вона має доступ та якими оперує. Суспільство має досить високі вимоги до безпеки системи в цілому, ажде глибока інтеграція її в людське життя із прогалинами по безпеці може поставити у майбутньому під удар не лише самого користувача системи.

Ще однією важливою рисою сучасного "розумного будинку" є високий рівень абстракції. Тобто - легкість налаштування та користування пересічною людиною і відсутність потреби у спеціальних знаннях та вміннях (що присутня, наприклад, при встановленні системи із протоколом X10).

					ІАЛЦ.467200.003 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Це може бути зручний користувацький інтерфейс, або ж наявність голосових команд, як, скажімо, у системі “Google Home”, що значно спрощує керування системою, зводячи його до простих розмов із системою (що не завжди користувача розуміє, будемо відверті). Система, загалом, має бути готова до роботи “із коробки”. “Дістав - ввімкнув - користуєшся” - мрія людини, що планує придбати та користуватися подібною системою.

Останнім, але дуже важливим пунктом є масштабованість системи та наявність великого банку модулів для неї. Тобто, якщо користувачу потрібно розширити систему, він може піти і придбати необхідний модуль, увімкнути його вдома і одразу ж користуватися ним. Користувачу не потрібна система, що може лише відчиняти двері, і більш нічого, та інша система, що керує лише світлом й не може бути поєднана із системою відкривання дверей. Це до речі, є однією із найскладніших проблем сьогодення. Така фрагментація призводить до купи незручностей і може не полегшувати людське життя, а зовсім навпаки, його ускладнювати та значно перевантажувати, адже за кожною із систем потрібен окремий контроль та налаштування, що також віднімає дорогоцінний час.

Отже, розглянемо доступні варіанти існуючих зараз на ринку систем та порівняємо їх, аби мати змогу на основі цих даних вибудувати власну ідею щодо створення системи.

1.3 Огляд існуючих систем

На даний момент ринок девайсів для “розумного будинку” досить багатий, проте, як було зазначено раніше, на жаль, більшість із систем, що представлені на ньому, автоматизують зовсім невелику кількість завдань, а тому не будуть розглянуті надалі, оскільки, скажімо ту саму систему Ajax досить важко віднести до “розумного будинку”, адже вона є просто дуже розвиненою системою безпеки.

Система Broadlink: створена переважно для гнучкого керування

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

побутовою електронікою, проте має дещо ширший функціонал, тож може взаємодіяти із широким асортиментом систем будинку. Вона швидко встановлюється і не потребує тривалого налаштування, має широкий асортимент датчиків (вологи, температури, освітленості, шуму, забруднення повітря і тд.), бездротова взаємодія пристроїв між собою, контроль по Wi-Fi, можливість простого масштабування.

Система Fibaro: відноситься до професійного сегменту, тож не є дружнім до користувача і потребує складного налаштування майстром при встановленні. Має великий банк датчиків, сценарії для використання, сумісність із Google Nest. Проте має високу вартість, обов'язкове підключення хабу через LAN, необхідність встановлення ПЗ на ПК.

Система Orvibo: недорогий і простий у експлуатації варіант, має мобільний додаток, автоматичне підключення сенсорів до Хабу, можливість масштабування системи, великий банк датчиків і можливість взаємодії із датчиками інших виробників, сценарії роботи, обов'язкове підключення хабу по LAN.

Система Xiaomi: повна автономність модулів, можливість масштабування, зручний інтерфейс керування через мобільний додаток, наявність сценаріїв, непоганий банк пристроїв.

Таблиця 1.1 - Порівняння найпопулярніших існуючих систем

Назва	Простота встановлення	Простота налаштування	Великий банк датчиків	Бездротовість	Можливість масштабування	Наявність сценаріїв	Інтерфейс керування	Взаємодія із датчиками і інших виробників	Голосове керування	Ціна
Broadlink	+	+	10 типів пристроїв	+	+	-	Веб-інтерфейс	-	-	від 70\$
Fibaro	-	-	14 типів пристроїв	Хаб підключається через LAN	-	+	Програма на ПК, мобільний додаток	+	+	від 285\$
Orvibo	+	+	14 типів пристроїв	Хаб підключається через LAN	+	+	Мобільний додаток	+	-	від 110\$

Продовження таблиці 1.1

Назва	Простота встановлення	Простота налаштування	Великий банк датчиків	Бездротовість	Можливість масштабування	Наявність сценаріїв	Інтерфейс керування	Взаємодія із датчиками і інших виробників	Голосове керування	Ціна
Xiaomi	+	+	більш ніж 19 типів пристроїв	+	+	+	Мобільний додаток	-	-	від 100\$
Google Nest	+	+	більш ніж 13 типів пристроїв	+	+	+	Голосовий асистент, мобільний додаток, інтерфейс хабу	+	+	від 100\$

Кінець таблиці 1.1

Система Google Nest: до останнього часу Google Nest був лише хабом із голосовою системою керування, що базується на Google Assistant, проте тепер обзавівся непоганим комплектом датчиків, що покривають досить велику кількість задач, а також є сумісність із купою інших пристроїв сторонніх виробників та можливість простого масштабування. Голосове керування та інтегрованість до інших продуктів корпорації робить цю систему дуже зручною у користуванні.

Отже, з огляду на найпопулярніші системи, що представлені зараз на ринку, очевидними критеріями для розроблюваної системи є масштабованість, зручний інтерфейс, наявність сценаріїв автоматизації. До того ж можна помітити, що жодна із систем не має жодних зручних способів взаємодії із людьми з обмеженими можливостями, особливо ті, що керуються голосом, адже, як зазначалося раніше, “розумний будинок” може значно спростити життя людям із обмеженими можливостями, яким не так легко дається побут, проте людям із вадами мови та слуху буде надзвичайно важко користуватися системою із високим рівнем абстракції, тобто голосовим керуванням, а система без цього не може похвалитися такою зручністю у користуванні. Тому, напевно, при розробці варто зацентруватися на інклюзивності системи.

Висновок до розділу 1

У першому розділі було розглянуто історичну генезу “розумного дому” та “інтернету речей”, а також сучасні тенденції цієї галузі і найяскравіших представників на сучасному ринку. Були зроблені висновки щодо характерних рис систем “розумного будинку”, а також напрямків перспективного розвитку, і, щонайголовніше, – повної порожнечі сектору систем у сфері інклюзивності, через що було обрано саме цей варіант системи із рішенням назвати таку систему “тихою”, що вказує на її призначення для людей із порушенням мовлення та слуху і покликано спростити їх життя, при цьому не прибігаючи до голосових команд системі.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

РОЗДІЛ 2 ПІДБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЗАДАЧІ

2.1 Особливості розробки ідеї

Отже, перейдемо до аналізу апаратної реалізації проєкту. Розглянемо широко вживані у подібних системах модулі та компоненти, а також ті компоненти, що можуть задовольняти вимоги та принципи для побудови необхідної системи, і, у ході порівняння, прийдемо висновку щодо того, яку апаратну складову слід використовувати при побудові проєкту. При виборі компонентів для реалізації системи слід керуватися такими принципами як:

- Економічна вигода - вартість виготовлення одного екземпляру не має переважати вартість вже готового існуючого рішення.
- Достатність обчислювальних можливостей для реалізації системи.
- Наявність достатньої кількості способів підключення периферії.
- Сучасність та підтримуваність компонентів - компоненти мають бути не застарілими технічно та мати довгострокову підтримку виробником чи, хоча б, ентузіастами.
- Наявність можливості бездротового з'єднання як одного із ключових аспектів сучасних тенденцій систем “розумного будинку” та “IoT”.
- Простота у роботі із компонентами.

У розроблюваній системі центральним елементом, як було виділено вище, стане хаб, що буде обробляти інформацію з інших пристроїв та керувати ними. Цей вузол буде побудований на мікроконтролері або ж мікрокомп'ютері, що відповідатиме за зв'язок із іншими пристроями, обробку даних та керування, а отже буде достатньо навантажений обчисленнями. На даний момент на ринку представлено безліч різноманітних мікроконтролерів із абсолютно різноманітними можливостями, від найбазовіших та

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

найпростіших, до найбільш широко функціональних і до мікрокомп'ютерів, що можуть обробляти велику кількість інформації і мають непересічні обчислювальні потужності. Розглянемо деякі із існуючих компонентів, та, у ході їх вивчення і порівняння можливостей та наявних модулів, визначимося із апаратною реалізацією проєкту.

2.2 Вибір апаратної платформи хабу

2.2.1 Платформа Arduino

Arduino є програмно-апаратною обчислювальною платформою для конструювання та прототипування простих систем, моделей в області електроніки, автоматики та робототехніки, що складається із основних компонентів: плати мікроконтролера з елементами вводу/виводу та середовище розробки на мові програмування високого рівня Arduino C, що є спрощеним варіантом C++. Плата побудована на мікроконтролері AVR Atmel із обов'язкою для інтеграції та програмування (кварцовий резонатор тактування 8МГц або 16МГц, лінійні стабілізатори напруг +5В або +3.3В тощо). Уся інформація про плату перебуває у відкритому доступі, а тому може бути використана будь-ким, хто хоче створювати плату власноруч. Відсутня потреба у зовнішньому програматорі через наявність записаного завантажувача (bootloader), що приємно полегшує роботу із платформою і дозволяє легко завантажувати програми на енергонезалежну пам'ять плати. Завантаження програм на плату відбувається за допомогою підключення її до комп'ютера через інтерфейс USB, або ж UART. Усі плати "сімейства" Arduino спроектовані із урахуванням можливості простого розширення їх функціоналу за допомогою "shields" (тобто плат розширення, "шилдів", "щитів"). Плати Arduino дають змогу користуватися значною кількістю виводів мікроконтролера у якості вхідних/вихідних контактів у зовні. Скажімо, плата Decimila дає доступ до 16 вхідних/вихідних контактів, 6 із них можуть генерувати широтно-імпульсно модульований сигнал, а 6 - бути

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

аналоговими входами. Вони доступні, зазвичай, на платі через штифтові роз'єми чи контактні площадки. Через них зазвичай підключаються “shields”. Лінійка Arduino має достатньо широкий асортимент плат, існують плати зі зменшеними чи спеціальними форм-факторами (Nano, Lilypad), існує велика кількість готових плат розширення. Також на ринку представлена вражаюча кількість модулів та електромеханічних елементів, що є сумісними із платами Arduino. Це призводить до значної популярності платформи та можливості реалізації на ній найрізноманітніших ідей через наявність необхідного для цього різноманітного апаратного інструментарію.

Нижче представлена порівняльна таблиця із характеристиками найпоширеніших плат Arduino, що представлені зараз на ринку.

Таблиця 2.1 — Порівняння існуючих моделей Arduino

Модель	Мікроконтролер	Флеш - пам'ять	EEPROM	SRAM	Двійкові входи/виходи	Двійкові входи/виходи з ШІМ	Аналогові входи	Інтерфейси	Габаритні розміри, мм	Живлення
ADK	ATmega2560	256 КБ	4 КБ	8 КБ	54	14	16	MAX3421E, USB хост, USB	101.6 × 53.3	5 В
BT	ATmega328P	32 КБ	1 КБ	2 КБ	14	4	6	Bluegiga WT11 Bluetooth	81.2 × 53.3	5 В
Diecimila	ATmega168	16 КБ	0.5 КБ	1 КБ	14	6	6	USB	68.6 × 53.3	5 В
Due	ATMEL SAM3X8E	256 КБ	0 КБ	50 КБ	54	16	16	UART, USB, SPI, JTAG	101.6 × 53.3	3.3 В
Duemilanove	ATmega168/328P	16/32 КБ	1 КБ	0.5 КБ	14	6	6	USB	68.6 × 53.3	5 В
Ethernet	ATmega328	32 КБ	1 КБ	2 КБ	14	4	6	Wiznet, Ethernet, MicroSD	68.6 × 53.3	5 В
Fio	ATmega328P	32 КБ	1 КБ	2 КБ	14	6	8	-	40.6 × 27.9	3.3 В
Leonardo	ATmega32u4	32 КБ	1 КБ	2 КБ	14	6	12	USB	68.6 × 53.3	5 В
LilyPad	ATmega168V або 328V	16 КБ	0.5 КБ	1 КБ	14	6	6	-	50 × 50	2.7 - 5.5 В

Продовження таблиці 2.1

Модель	Мікроконтролер	Флеш-пам'ять	EEPROM	SRAM	Двійкові входи/виходи	Двійкові входи/виходи з ШІМ	Аналогові входи	Інтерфейси	Габаритні розміри, мм	Живлення
Mega	ATmega1280	128 КБ	4 КБ	8 КБ	54	14	16	USB	101.6 × 53.3	5 В
Mega 2560	ATmega2560	256 КБ	4 КБ	8 КБ	54	14	16	USB	101.6 × 53.3	5 В
Nano	ATmega168 або 328	16/32 КБ	1 КБ	0.5 КБ	14	6	8	USB	43 × 18	5 В
Pro Mini	ATmega328P	32 КБ	1 КБ	2 КБ	14	6	8	UART, USB	33 × 18	3.3 - 5 В
Uno	ATmega328P	32 КБ	1 КБ	2 КБ	14	6	6	USB	68.6 × 53.3	5 В

Кінець таблиці 2.1

Проте асортимент існуючих плат не обмежується лише тими, що представлені вище, існують і інші версії плат, що мають деякі удосконалення, як, скажімо, наявність додаткових виходів чи вбудованого Wi-Fi модулю, і їх також варто розглянути, адже одним із критеріїв вибору плати, що зазначено на початку розділу, є бездротовість, а, як видно із порівняльної таблиці, бездротове з'єднання наявне у, так би мовити, “базовій комплектації”, лише у моделі BT, що має інтерфейс Bluetooth.



Рис 2.1 Arduino BT — Вид згори [1]

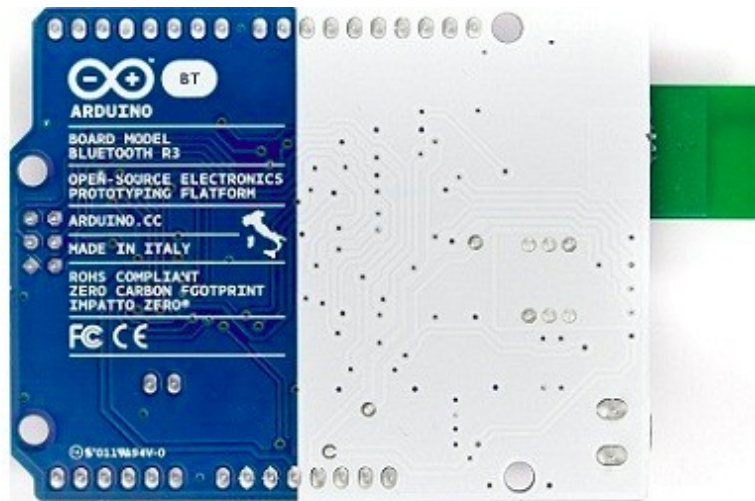


Рис 2.2 Arduino BT — Вид знизу [1]

Вона базується на модулі Bluegiga WT11 bluetooth, що може працювати в режимах приймача і передавача, на частоті 2.4ГГц. Модуль використовує протокол Bluetooth 2.0 EDR Class 1, має швидкість передачі даних до 3 Мбіт/сек, а також обладнаний як вбудованою антеною, так і роз'ємом UFL для підключення зовнішньої антени.

Із, так би мовити, “розширених версій” варто розглянути плату Arduino Uno Wi-Fi, що є версією плати Arduino Uno із встановленим “із коробки” Wi-Fi [3] модулем для бездротового зв’язку. Як заявляє виробник: “Arduino Uno Wi-Fi є найлегшим способом доступу до IoT зі стандартним форм-фактором сімейства Uno” [4]. За бездротове з’єднання у даній платі відповідає чип u-blox NINA-W102, він надає можливість не тільки підключати плату до інших пристроїв за допомогою Wi-Fi 802.11b/g/n на частоті в 2.4ГГц, а також використовуючи Bluetooth v4.2 [18]. Виробник у документації до чипу прямим текстом зазначає, що дане рішення було розроблено спеціально із поглядом на концепцію “інтернету речей” і для використання у ній [5].

					ІАЛЦ.467200.003 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

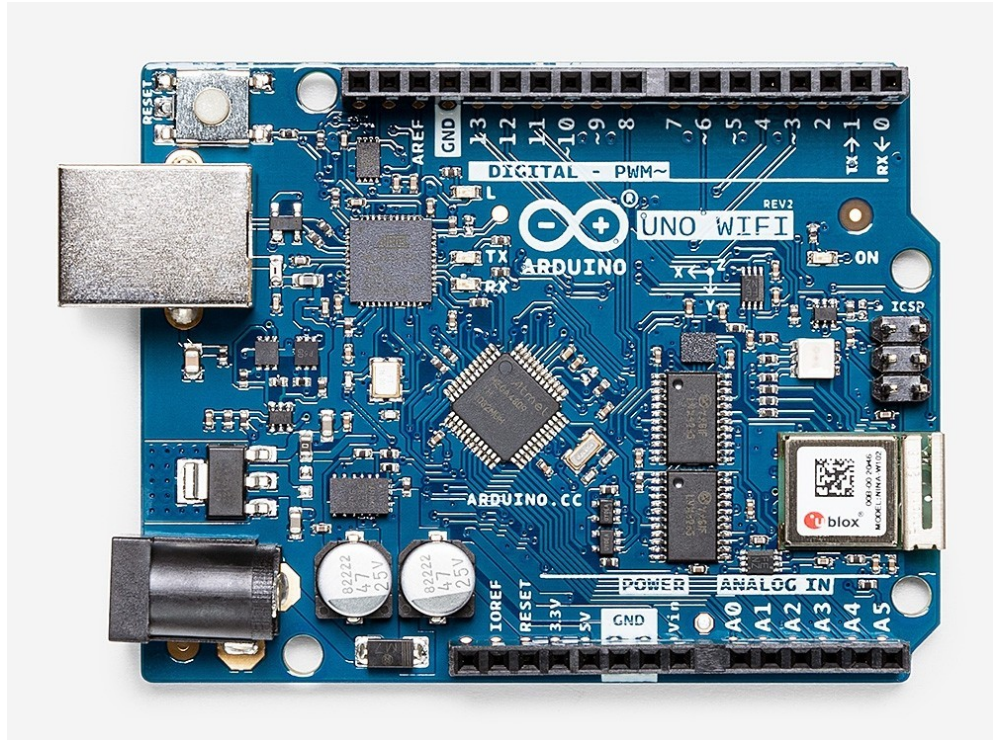


Рис 2.3 Arduino Uno Wi-Fi — Вид згори [3]

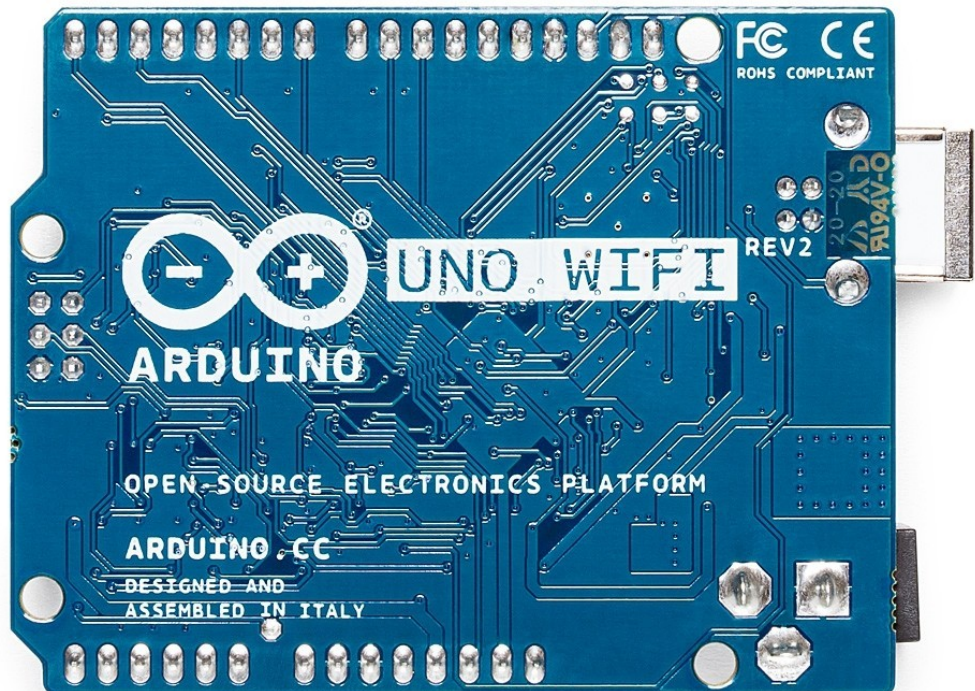


Рис 2.4 Arduino Uno Wi-Fi — Вид знизу [3]

Змн.	Арк.	№ докум.	Підпис	Дата

Також вартою уваги буде плата Arduino Nano 33 IoT [2], що прямо своєю назвою натякає на сферу її використання (принаймні, із точки зору розробників). Це, так само як і Arduino Uno Wi-Fi, є “розширеною версією” плати Arduino Nano, що її було додатково обладнано таким самим чипом бездротового зв’язку u-blox NINA-W102. Також, іншою важливою відмінністю від звичайної плати Arduino Nano, є використання у ній мікроконтролера SAMD21 Cortex-M0.

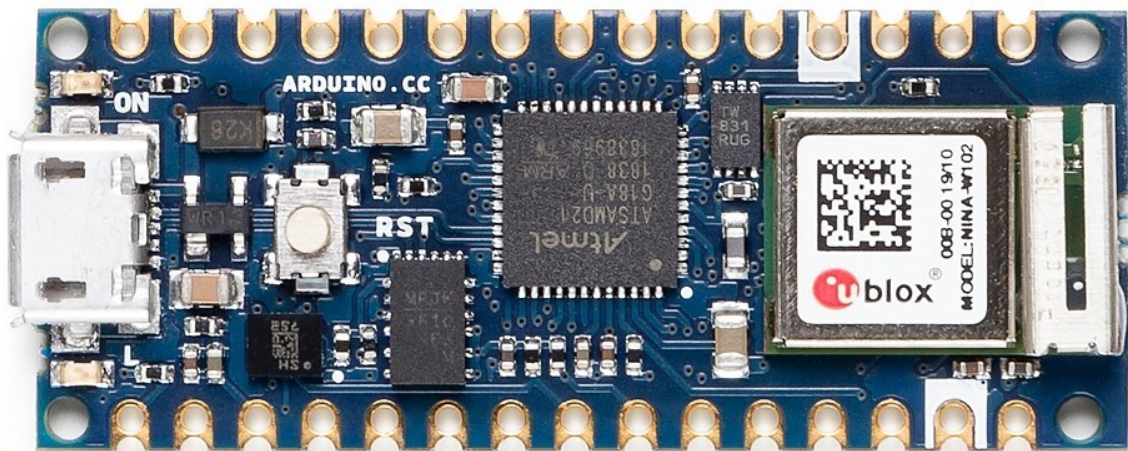


Рис 2.5 Arduino Nano 33 Iot — Вид згори [2]

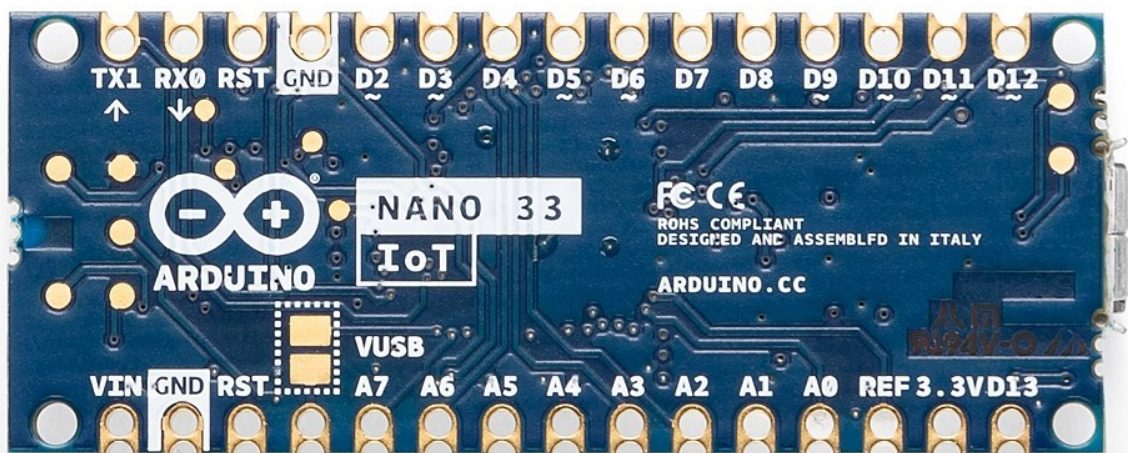


Рис 2.6 Arduino Nano 33 Iot — Вид знизу [2]

Проте, не варто зосереджувати усю увагу тільки на готових рішеннях із бездротовим з'єднанням, оскільки практично будь-яка плата Arduino може бути надзвичайно легко дообладнана таким рішенням, як Arduino Wi-Fi Shield, або Wi-Fi модулем із сімейства ESP, що надзвичайно добре зарекомендували себе у середовищі розробників проєктів під Arduino, як із точки зору роботи, так і з економічної, оскільки часто плати, дообладнані даними модулями мають меншу собівартість за готові плати по типу Arduino Nano 33 IoT, чи Arduino Uno Wi-Fi. Отже, з огляду на специфіку задачі, варто розглянути можливості реалізації проєкту на Arduino Uno Wi-Fi чи Arduino Uno із Wi-Fi Shield чи додатковим модулем від ESP, як найбільш зручні, та підходящі у використанні варіанти.



Рис 2.7 Різновиди бездротових мікроконтролерів ESP [22]

2.2.2 Серія мікроконтролерів ESP

Наступним кандидатом на апаратну складову проекту варто розглянути серію мікроконтролерів ESP від виробника Espressif Systems. Мікроконтролери даного виробника стали популярними серед розробників відносно нещодавно - лише у 2014 році, із виходом на ринок перших продуктів, що базувалися на мікроконтролері ESP8266. Свою популярність вони здобули здебільшого через економічну складову - при усіх своїх можливостях та обчислювальних потужностях продукти на базі цього мікроконтролера мали дуже низьку ціну. ESP8266 це мікроконтролер із інтерфейсом Wi-Fi, що також може виконувати програми із зовнішньої флеш-пам'яті, під'єднаної за SPI інтерфейсом. Згодом компанія Espressif почала виробництво мікроконтролера ESP8285, що фактично є ESP8266 із вбудованою флеш-пам'яттю на 1 МБайт.

Мікроконтролер ESP8266 оснащено:

- 80 МГц 32-бітним процесором компанії Tensilica - Xtensa L106.
- 18 портами вводу/виводу, що з них лише 11 можна використати.
- Інтерфейси: SPI, I²C, I²S, UART.
- 10-бітний аналогово-цифровий перетворювач.
- Стандарт комунікації в локальній бездротовій мережевій зоні IEEE 802.11 b/g/n Wi-Fi.
- Має підтримку WEP та WPA/WPA2. [9]

Як видно із специфікацій, мікроконтролер не оснащено енергонезалежною пам'яттю, для використання користувачем, програма виконується із зовнішньої пам'яті, шляхом динамічного завантаження потрібних частин програми у кеш інструкцій. Також, виробником не надається документації до внутрішньої периферії контролера, проте надається набір бібліотек, за допомогою яких програміст має можливість отримати доступ до периферії. Ці бібліотеки активно використовують

					ІАЛЦ.467200.003 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

оперативну пам'ять контролера, але виробник не надає відомості про точну її кількість, а лише приблизно оцінює її кількість, що залишається вільною після лінування бібліотек, у 50 кБайт. Проте ентузіастами було досліджено, що ESP8266 має 32 кБайт кешу інструкцій та 80 кБайт пам'яті даних.



Рис 2.8 Модуль на базі мікроконтролера ESP8266 [8]

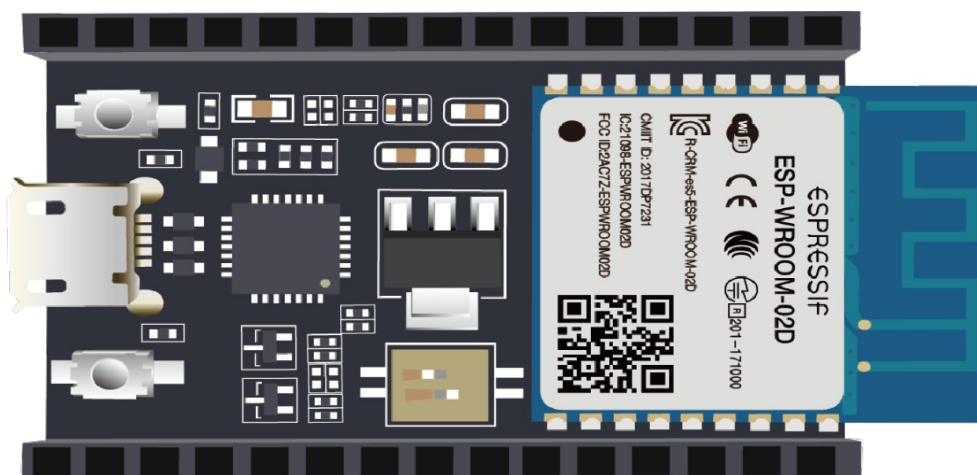


Рис 2.9 DevKit на базі модуля ESP-WROOM-02D [8]

У 2015 році Espressif модернізувала мікроконтролер ESP8266 і представила удосконалену серію лінійки - ESP32. ESP32 це серія дешевих, із низьким енергоспоживанням, мікроконтролерів “системи на кристалі” з інтегрованим Wi-Fi та дворежимним Bluetooth. Серія ESP32 використовує мікропроцесори Tensilica Xtensa LX6 в двох-ядерних та одноядерних варіаціях.

Основні характеристики серії ESP32:

- Xtensa dual-core (або single-core) 32-бітний LX6 мікропроцесор, із частотою 160 або 240 МГц.
- Ко-процесор ультра низького енергоспоживання.
- 520 Кб ОЗУ, 448 Кб ПЗУ. Зовнішні ОЗУ та ПЗУ підключаються по інтерфейсу SPI, є можливість підключення до 64 Мб.
- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR та BLE (використовує одну антену із Wi-Fi).
- Периферійні інтерфейси:
 - 12-бітний АЦП до 18 каналів,
 - 8-бітний 2х ЦАП,
 - 4х SPI,
 - 2х I²C,
 - 2х I²S,
 - 3х UART,
 - Контроллер карт пам'яті SD/SDIO/CE-ATA/MMC/eMMC,
 - CAN 2.0,
 - Інфрачервоний порт,
 - Ethernet MAC інтерфейс,
 - ШІМ модуляція для моторів та світлодіодів,
 - датчик ефекту Холла.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

- IEEE 802.11 із підтримкою WPA, WPA/WPA2 і WAPI.
- Таймер.
- Криптографічна апаратна акселерація: AES, SHA-2, RSA, ECC, RNG.

Не дарма вище було зазначено, що ESP32 являє собою саме серію пристроїв, адже компанією випускається декілька варіантів цього контролера. Нижче наведено порівняльну таблицю із основними відмінностями різних варіацій мікроконтролера цього сімейства.

Таблиця 2.2 — Порівняння різних варіацій мікроконтролера ESP32

Модель	Мікропроцесор	Пам'ять	Бездротове з'єднання	Додаткові інтерфейси
ESP32-S2	Одноядерний Xtensa LX7 CPU до 240 МГц. До 64 FLOPS за цикл.	320 Кб SRAM, 128 Кб ROM, і 16 Кб RTC	без Bluetooth	USB OTG
ESP32-S3	Двоядерний Xtensa LX7 CPU до 240 МГц. Додані інструкції для прискорення застосунків із машинним навчанням	384 Кб RAM, додаткові 384 Кб SRAM	Bluetooth 5 and LE	USB OTG, 44 програмовані GPIO
ESP32-S6	Одноядерний 32-бітний RISC-V CPU до 160 MHz	400 Кб of SRAM і 384 Кб ROM	Bluetooth 5 and LE, Wi-Fi 6	22 програмовані GPIO
ESP32-C3	Одноядерний 32-бітний RISC-V CPU до 160 MHz	401 Кб of SRAM і 384 Кб ROM	Bluetooth 5 and LE	22 програмовані GPIO, контактна сумісність із ESP8266

Судячи із інформації на сайті [10] виробника, на даний момент підтримка та виробництво моделі ESP32-S6 припинена.

Окрім цього, виробником пропонуються не тільки самі мікроконтролери, але й так звані DevKit для більш зручного прототипування та взаємодії із іншими приладами. Вони мають додаткову флеш-пам'ять, кнопки керування та індикатори стану, можуть мати додаткові входи/виходи, дисплеї, елементи живлення тощо. Одним із найбільш цікавих для даного проєкту DevKit'ом є ESP32-DevKitC, що прямо "із коробки" має сумісність із

2.2.3 Мікрокомп'ютери Raspberry Pi

Raspberry Pi це одноплатні комп'ютери, що розробляються фондом Raspberry Pi Foundation із Британії в асоціації із Broadcom. Початковою метою проекту Raspberry Pi було сприяння вивченню базису комп'ютерних наук у школах та країнах, що розвиваються. Проте оригінальна модель стала набагато популярнішою, аніж цього очікувалося, і знайшла своє застосування за межами початкового цільового ринку. На даний момент мікрокомп'ютери цього бренду широко використовуються у різноманітних системах та сферах [19], таких як робототехніка, автоматизація, моніторинг погоди тощо, перш за все через свою низьку ціну, модульність та дизайн. Також її досить любляють комп'ютерні та електротехнічні хобісти, через наявність інтерфейсів HDMI та USB.

Raspberry Pi базується на “системі на чипі” Broadcom BCM2835, що включає у себе процесор ARM із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 чи 256 Мбайт оперативної пам'яті. Енергонезалежна пам'ять, вбудована, відсутня, проте її компенсує наявність слоту для карт пам'яті.

Через значну популярність мікрокомп'ютера, а також його заснованість на ARM архітектурі, була випущена досить велика кількість програмного забезпечення для роботи із даними мікрокомп'ютерами, зокрема наступні операційні системи:

- Raspbian OS - Debian-подібний дистрибутив Linux.
- Pidora - варіант дистрибутиву Fedora виконаний спеціально під Raspberry Pi
- RISC OS
- Windows 10 ARM64 - варіація Windows 10 для архітектури ARM
- Windows 10 IoT Core
- Plan 9

					ІАЛЦ.467200.003 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- FreeBSD
- OpenBSD
- Ubuntu 20.04 Focal Fossa та Ubuntu 20.10 Groovy Gorilla від Canonical офіційно підтримують роботу на Raspberry Pi

Починаючи із 2014 року фундація випускає різноманітні варіації мікрокомп'ютера із метою задовольнити різноманітні потреби розробників, що користуються даною платформою для своїх проєктів. Різні моделі відрізняються між собою здебільшого кількістю та варіантами інтерфейсів підключення зовнішніх пристроїв, а також форм-фактором, і, звісно ж, ціною. На момент написання роботи, на веб-сайті фундації представлено 11 варіантів мікрокомп'ютера Raspberry Pi: 400 Unit, Pico, 4 Model B, 3 Model A+, 3 Model B+, 3 Model B, 2 Model B, 1 Model B+, 1 Model A+, Zero W, Zero. Із них, для даного проєкту, точно не потрібно розглядати модель Raspberry Pi 400 Unit, оскільки вона має у собі інтегровану виробником клавіатуру та передвстановлену операційну систему Raspberry Pi OS (Raspbian OS), що негативно сказується на гнучкість, у виборі засобів програмування, а також робить цей мікрокомп'ютер надто громіздким для зручного використання. Задля порівняння інших, актуальних на даний момент моделей мікрокомп'ютера Raspberry Pi скористаємося наступною порівняльною Таблицею 2.3:

Таблиця 2.3 — Порівняння мікрокомп'ютерів сімейства Raspberry Pi

Модель	Кристал	К-ть ядер	Тактова частота	RAM	USB	Ethernet	Video Out	Audio Out	GPIO	Бездротовий зв'язок	Розмір
Raspberry Pi 4 Model B	Broadcom BCM2711 Cortex-A72 (ARM v8) 64-bit	4	1.5 GHz	2-8 GB	2x USB 3.0, 2x USB 2.0, USB-C OTG	Gigabit	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	Wi-Fi 2.4GHz та 5GHz 802.11 b/g/n/ac, Bluetooth 5.0	56мм x 85мм

Продовження таблиці 2.3

Модель	Кристал	К-ть ядер	Тактова частота	RAM	USB	Ethernet	Video Out	Audio Out	GPIO	Бездротовий зв'язок	Розмір
Raspberry Pi 3 Model B+	Broadcom BCM2837 B0 Cortex-A53 64-bit	4	1.4 GHz	1 GB	4x USB 2.0	Gigabit - Over USB 2.0, PoE	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	Wi-Fi 2.4GHz та 5GHz 802.11 b/g/n/ac, Bluetooth 4.2, BLE	56мм x 85мм
Raspberry Pi 3 Model B	Broadcom BCM2837 ARMv8-A		1.2 GHz	1 GB	4x USB 2.0	Gigabit	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	Wi-Fi 2.4GHz та 5GHz 802.11 b/g/n/ac, Bluetooth 2.0/4.1	56мм x 85мм
Raspberry Pi 3 Model A+	Broadcom BCM2837 B Cortex-A53 64-bit	4	1.4 GHz	512 MB	1x USB 2.0	-	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	Wi-Fi 2.4GHz та 5GHz 802.11 b/g/n/ac, Bluetooth 4.2, BLE	56мм x 65мм
Raspberry Pi 2 Model B	Broadcom BCM2836 ARMv7-A	4	900 MHz	1 GB	4x USB 2.0	Gigabit	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	-	56мм x 85мм
Raspberry Pi 1 Model B+	Broadcom BCM2835 ARM1176 JZF-S	1	700 MHz	512 MB	4x USB 2.0	Gigabit	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	-	56мм x 85мм

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.3

Модель	Кристал	К-ть ядер	Тактова частота	RAM	USB	Ethernet	Video Out	Audio Out	GPIO	Бездротовий зв'язок	Розмір
Raspberry Pi 1 Model A+	Broadcom BCM2835 ARM1176 JZF-S	1	700 MHz	256 MB	1x USB 2.0	-	HDMI, аналоговий вихід схрещений із аудіо джеком	3.5mm jack	40 пінів	-	56мм x 65мм
Raspberry Pi Pico	RP2040 dual-core Arm Cortex-M0+	2	133 MHz	264 KB	1x USB 1.1 Device / Host	-	-	-	26 пінів	-	51мм x 21мм
Raspberry Pi Zero	Broadcom BCM2835 ARM1176 JZF-S	1	1 GHz	512 MB	1x micro OTG	-	HDMI, або через пін	через HDMI	40 пінів	-	30мм x 65мм
Raspberry Pi Zero W	Broadcom BCM2835 ARM1176 JZF-S	1	2 GHz	512 MB	1x micro OTG	-	HDMI, або через пін	через HDMI	40 пінів	Wi-Fi 2.4GHz та 5GHz 802.11 b/g/n/ac, Bluetooth 4.1, BLE	30мм x 65мм

Кінець таблиці 2.3

Також фундація Raspberry Pi виготовляє офіційні аксесуари для своїх мікрокомп'ютерів, у число яких входять також різноманітні плати розширення, що підключаються здебільшого за допомогою пінів GPIO, або ж, іноді, інших наявних інтерфейсів, і можуть значно розширювати функціонал мікрокомп'ютера. Плати розширення надають змогу вводу/виводу звуку (існують різноманітні зручні форм-фактори), також існують модулі - камери, телевізійні DVB-T2 антени, тачскріни. Також існують так звані "Compute Modules", що являють собою мікрокомп'ютери для використання у сфері еMBED. Вони мають зменшений форм-фактор і більшість із них обладнані

DDR2 SODIMM коннектором для зручної інсталяції за необхідністю користувача.

Проте плати розширення виготовляються не тільки власне фундацією Raspberry. На ринку на даний момент представлена велика кількість сумісних із Raspberry Pi плат розширення, чим треба завдячувати великій кількості пінів GPIO на усіх варіаціях мікрокомп'ютера, що дає змогу підключати до них велику кількість різноманітних пристроїв, чим і користуються виробники сторонньої периферії. Як і в платах розширення від самої фундації Raspberry, в асортименті плат від сторонніх розробників представлені і дисплеї, тачскріни, камери, проте, також, представлені і різноманітні модулі, що не входять в асортимент компанії, як то: плати забезпечення безперебійного живлення від акумуляторів, плати із додатковими роз'ємами (USB, Ethernet і т.д.), плати для роботи із мережами GSM, GPRS, радіочастотами, плати що збільшують кількість пінів GPIO і ще безліч різноманітних плат фактично на будь-який смак розробника. Дані плати надзвичайно сильно споріднюють мікрокомп'ютери Raspberry Pi із мікроконтролерами Arduino та шильдами, що використовуються для розширення їх можливостей.

Однією із плат розширення, яку варто ґрунтовно дослідити задля побудови даного проєкту є плата RaZberry (Рис. 2.11).

Ця плата розширення займає 10 пінів GPIO і дозволяє використовувати для передачі даних протокол Z-Wave, який, як було висновувано у першому розділі, наразі є одним із популярних протоколів зв'язку та керування для систем “розумного будинку”.

					ІАЛЦ.467200.003 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

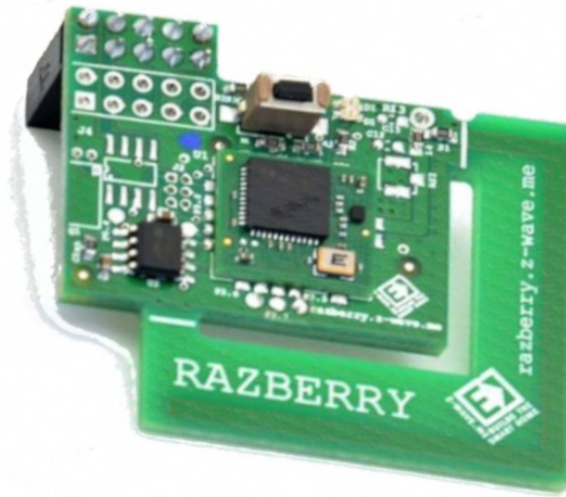


Рис 2.11 Плата розширення для роботи Raspberry Pi із протоколом Z-Wave

Використання розповсюдженого протоколу несе за собою чимало плюсів, адже із цим легко закривається гештальт масштабованості системи, бо до неї можна буде легко підключити інші пристрої, що використовують цей самий протокол і керувати ними, і відпадає потреба у подальшій розробці різноманітних модулів, чи то пак, актуаторів для системи, що будуть із нею працювати, і, як наслідок, такий підхід веде до зменшення (чи хоча б, не збільшення) фрагментації у концепції IoT, що, як було сказано у першому розділі, досить негативно впливає на розвиток цієї царини як такої.

Проте при найближчому розгляді цього, здавалося б, майже божественного варіанту, виявляється, що далеко не все так добре, як здається не тільки на перший, а й навіть на третій погляд. Насправді, уся “екосистема” Z-Wave є досить закритою. Так, разом із платою, що, до речі, коштує мало не як сам Raspberry Pi, користувачу надається доступ до Z-Wave API, що дає можливість розробляти свій інтерфейс користувача (також можна використовувати застосунки, що були розроблені авторами Z-Wave, проте вони надто прості, як на комерційний продукт), прописувати власноруч сценарії роботи, а комплектне програмне забезпечення динамічно керує

									Арк.
									37
Змн.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.467200.003 ПЗ				

Отже для того, аби побудувати систему “розумного будинку”, треба під’єднувати власний модуль до Arduino Nano із шильдом, що дозволить долучити його до мережі Z-Wave, керування якою відбувається за допомогою мікрокомп’ютера Raspberry Pi із платою розширення RaZberry (адже тільки для нього існує необхідне розробнику програмне забезпечення). А у купі із не надто великим банком пристроїв, представленим на ринку, що використовують даний протокол (це, здебільшого, розетки, вимикачі, реле та датчики диму/вологи/відкривання) та достатньо великою ціною, адже наведена вище зв’язка, що потрібна для підключення власного модуля до системи, обійдеться користувачу приблизно у 6000 гривень, що відповідає цінам готових комерційних систем, використання такої плати розширення цілковито втрачає сенс із, перш за все, економічної, а потім, і користувацької точки зору. Тому ідея інтегрування даного проєкту у “екосистему” бездротового протоколу Z-Wave, нажаль, зазнає фіаско через наведені причини, не дивлячись на низку позитивностей, які така інтеграція могла принести.

З огляду на поставлені на початку другого розділу критерії вибору апаратної частини, розумним буде виокремити із усього наявного асортименту мікрокомп’ютерів Raspberri Pi для подальшого розгляду лише ті, що мають бездротовий зв’язок, а проте, наявність в усіх них досить великої кількості зовнішніх пінів GPIO не робить цей аспект аж надто критичним, оскільки із їх допомогою можна підключити велику кількість різноманітної периферії, у тому числі і модулі бездротового зв’язку.

Виходячи із порівняльної таблиці, одним із найкращих варіантів з поміж мікрокомп’ютерів цієї фундації є Raspberry Pi Zero W (ілюстрації — Рис. 2.14-15), що від початку обладнаний необхідною периферією у вигляді інтерфейсів бездротового зв’язку, має достатні обчислювальні можливості, а також 40 пінів GPIO, що дає можливість для розширення

					ІАЛЦ.467200.003 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

можливостей системи у подальшому, а також простір для маневру при необхідності у додатковій периферії при розробці.

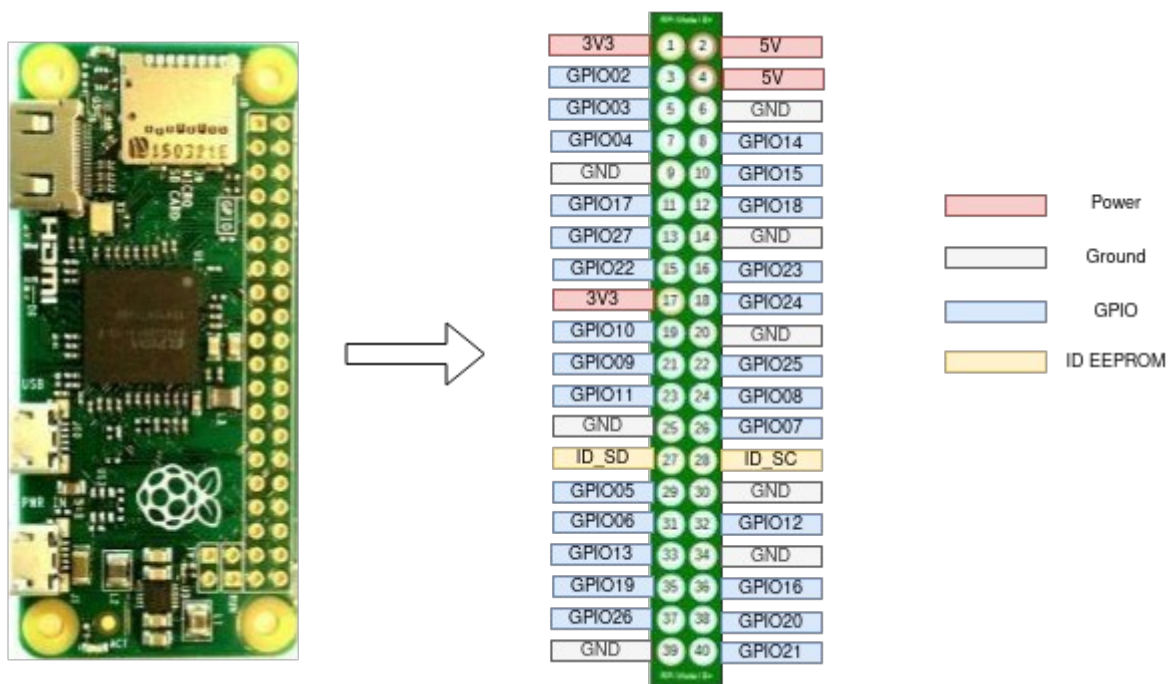


Рис 2.14 Raspberry Pi Zero W — розпіновка

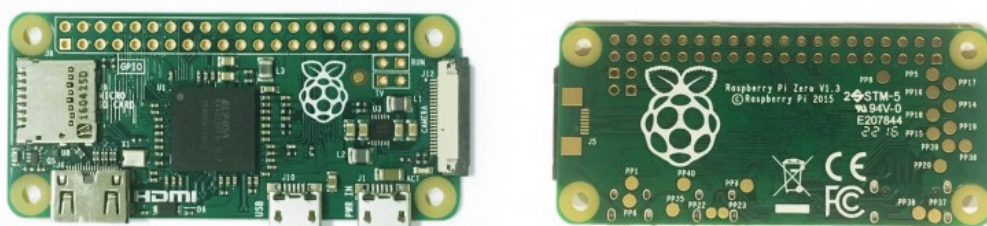


Рис 2.15 Raspberry Pi Zero W — зовнішній вигляд

Принадгідно варто зауважити, що ціна даної моделі мікрокомп'ютера досить помірнa.

Отже, в решті решт, з поміж наявних на ринку на даний момент рішень було обрано мікрокомп'ютер Raspberry Pi Zero WH, через ряд ключових відмінностей, що вигідно його виділяють з поміж інших контролерів, розглянутих у другому розділі даної роботи. Цей мікрокомп'ютер має одні з найбільш компактних розмірів серед претендентів, усього 30мм x 65мм, при цьому достатньо потужний для того, аби запускати

не просто програмний код, що було загружено на нього через програматор. а повноцінну операційну систему, що наразі не є, звісно, критичним фактором, проте в подальшій розробці буде дуже доречною річчю. Raspberry Pi Zero WH має достатні обчислювальні ресурси не тільки для задач, що будуть покладені на нього у межах даної роботи, а й великий потенціал для подальшого доопрацювання та розширення системи. До того ж, він обладнаний сучасними бездротовими інтерфейсами, що дозволяє їх широко використовувати в межах системи та відсуває необхідність у пошуку додаткових рішень для бездротовості, що, нагадаємо, є одним із основних пунктів у розробці сучасної системи “розумного будинку”. Мікрокомп’ютер виконує критерії економічної вигоди, що були поставлені на початку цього розділу роботи, адже коштує близько 25 доларів, що звісно дорожче за деякі контролери, що були розглянуті вище, проте не надто суттєво, особливо із огляду на обчислювальні можливості. Також виконується умова достатньої кількості способів підключення периферії, адже Raspberry Pi Zero WH має і бездротові з’єднання, і USB (докладніше - див. Таблицю 2.3), і, щонайголовніше, 40 пінів GPIO, до яких можна підключити різноманітні пристрої. Також обраний мікрокомп’ютер є досить сучасним і буде ще досить довгий час підтримуватися виробником, що гарантує постійні оновлення програмного забезпечення і іншу підтримку. Raspberry Pi Zero WH підтримує велику кількість різноманітних програмних рішень та мов програмування, отже під час розробки системи із використанням цього мікрокомп’ютера можна у повній мірі буде використати арсенал знань, набутих за період навчання, і з великою ефективністю їх застосувати на практиці. Таким чином усі перелічені вище риси роблять мікрокомп’ютер Raspberry Pi Zero WH надзвичайно гарним вибором для розробки системи “тихого розумного будинку”, і тому на ньому ми і зупинимося.

					ІАЛЦ.467200.003 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3 Вибір RFID ридера

Другим за важливістю елементом системи, що її реалізацію на меті має даний проєкт, є модуль безконтактного зчитування або ж радіочастотної ідентифікації, тобто RFID. Саме використанням цього модулю система завдячує своєю “тихістю”. Тобто, замість використання голосових команд, події у системі починаються при піднесенні мітки до модуля радіочастотної ідентифікації. За задумом даний модуль розташовується усередині дверних ручок будинку (або поруч із ними), а безконтактна мітка є інсталюваною у кільце на руці користувача. Таким чином, під час відкривання дверей до будинку (або ж міжкімнатних дверей) само собою відбувається зчитування мітки із кільця користувача, і запуск події у системі. Тому до підбору модуля RFID варто підійти також ґрунтовно.

Після дослідження наявних на ринку RFID модулів вибір пав на модуль RC522 від виробника RobotDyn (рис. 2.16-17). Він заснований на досить популярній мікросхемі MFRC522, що має добре реноме у середовищі розробників, проте також присутні досить суттєві відмінності від інших подібних йому модулів. Основною відмінністю є форм-фактор даного модуля, тобто його компактні розміри, що становлять усього 36мм x 36мм. При таких компактних розмірах увесь функціонал було збережено в повному обсязі і заявлені виробником характеристики повністю задовольняють розробку системи цього проєкту.

Основні характеристики модуля:

- Напруга живлення: 3.3В - 5В
- Струм живлення: 13-26мА
- Струм режиму очікування: 10-13мА
- Струм сплячого режиму: < 80мкА
- Дальність зчитування до 6 см, що є достатнім для можливих сценаріїв установки цього модуля

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

- Робоча частота: 13.56МГц - одна із найпопулярніших частот для роботи із безконтактними чипами
- Швидкість передачі даних по інтерфейсу SPI 10 Мбіт/с

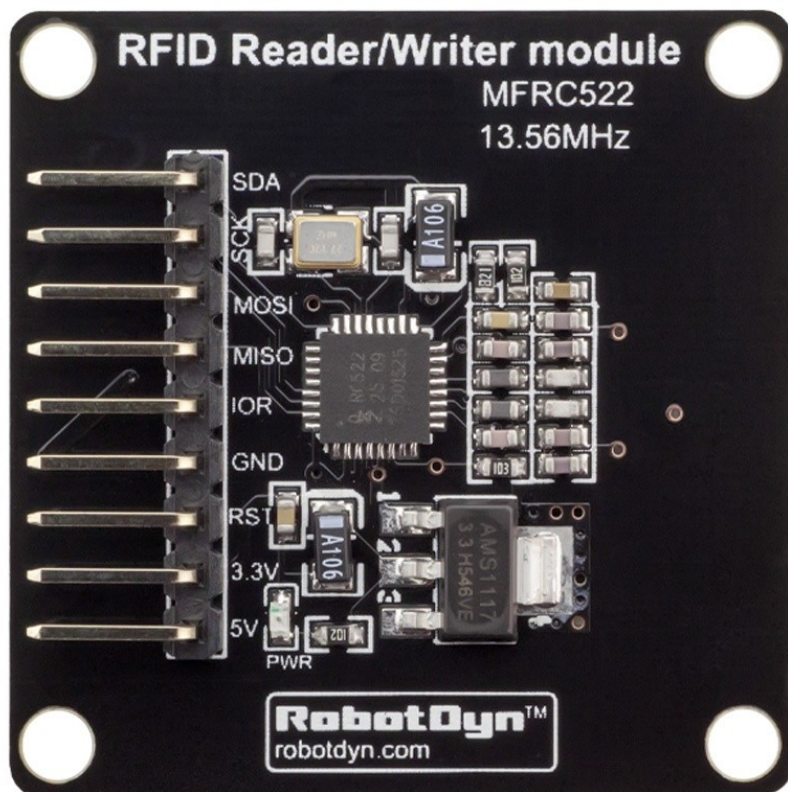


Рис 2.16 RFID Reader RC522 зовнішній вигляд

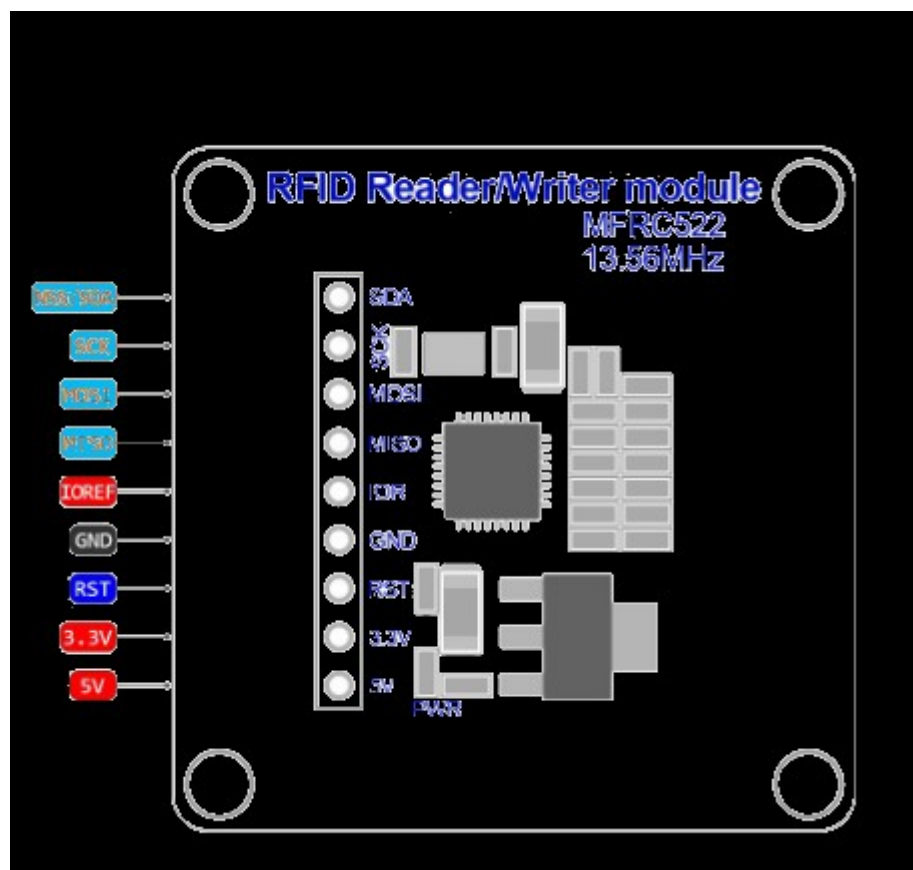


Рис 2.17 RFID Reader RC522 схема

Проте постає питання підключення обраного RFID ридера до хабу, що його у даній системі уособлює обраний вище Raspberry Pi Zero W. З огляду на те, що система планує оперувати не лише одним подібним модулем виключено варіант прямого з'єднання ридера із хабом, оскільки важко передбачити як себе вестимуть керуючі сигнали при передачі дротовим з'єднанням на суттєві відстані при розташуванні системи у великому будинку, також тоді вартує розробити захист та фільтрацію паразитних струмів, що можуть виникати при деякій довжині сигнальної лінії тощо. А найголовніше - дротове з'єднання модулів системи із її центром значно обтяжує користувацький досвід при монтажі та використанні системи. Тому резонно буде реалізувати бездротовий зв'язок модуля RFID ридера із хабом.

2.4 Вибір мікроконтролера для ридера

Для вибору мікроконтролера, що буде здійснювати бездротовий

					ІАЛЦ.467200.003 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

зв'язок та керування модулем скористаємося даними дослідження, що були здобуті на початку другого розділу. З огляду на низьке навантаження даного вузла, а також його місцезнаходження, слід звернути особливу увагу при виборі на критерії:

- компактності, адже модуль ридера обирався саме із суджень компактності задля простого інсталювання у будинку цього елемента системи, тож і контроллер, що передаватиме дані від ридера має бути компактним;
- наявності інтерфейсу SPI. Ридер передає дані за даним інтерфейсом із досить високою швидкістю, тому, хоча за допомогою простих маніпуляцій його і можна перевести на інші інтерфейси (UART, до прикладу), проте саме SPI забезпечуватиме гарну швидкодію;
- низького енергоспоживання, адже ридер із контролером встановлюватиметься у двері біля дверної ручки, а отже питання живлення є також досить важливим;
- ціна, адже для кращої автоматизації дані модулі мають бути встановлені на більшість дверей будинку, і тому собівартість одного модуля варто якомога значніше знизити.

З огляду на ці критерії та дані, отримані на початку розділу, кращим вибором із мікроконтролерів видається дещо вже застарілий, проте усе ще підтримуваний, дешевий та надійний ESP8266 (Рис. 2.18), що його можливостей буде достатньо для реалізації бездротової пересилки даних, отриманих від RFID ридера за протоколом SPI, а наявність вільних пінів GPIO, що залишаться після підключення його до мікроконтролера, дозволять виводити з їх допомогою користувачу корисну інформацію, як то повідомлення про помилки, тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

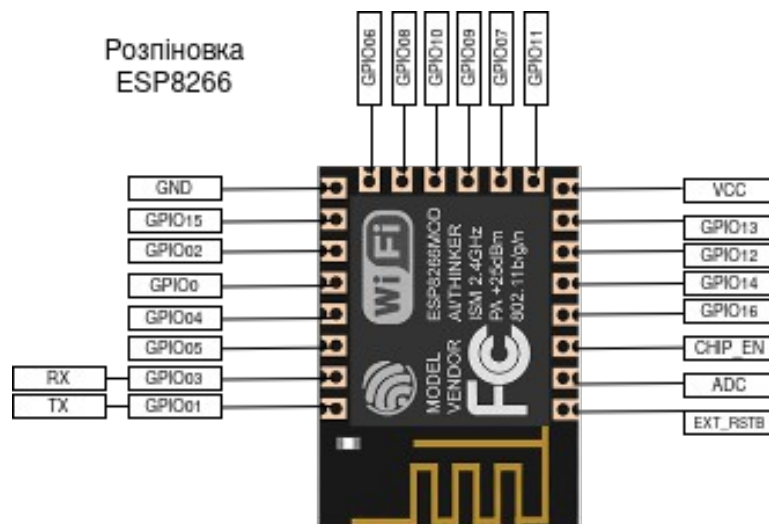


Рис 2.18 Розпіновка ESP8266

Отже з огляду на поставлені до обладнання на початку розділу вимог та у ході порівняння наявного на даний момент вибору, доступного для пересічного розробника, з об'єктивних, неодноразово вище перелічених причин для реалізації апаратної складової проекту було обрано наступну апаратуру:

- Хаб - Raspberry Pi Zero W
- Модуль RFID - RobotDyn RC522
- Мікроконтролер для комутації Модуля RFID та хабу - ESP8266

Висновок до розділу 2

У другому розділі даної роботи було висунуто конкретні вимоги до обладнання для апаратної реалізації проєкту “системи тихого розумного будинку” та остаточно сформовано концепцію майбутньої системи. Було проведено ґрунтовний аналіз наявних на ринку рішень у сфері мікроконтролерів, мікрокомп’ютерів та інших обчислювальних систем, модулів безконтактного зчитування RFID міток, в ході якого з’ясовано сильні та слабкі сторони доступного для пересічного розробника на ринку, на момент написання, обладнання, та обрано найвигідніший, та більш слушний апаратний базис для реалізації усіх складових проєкту даної роботи.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

РОЗДІЛ 3 ОПИС РЕАЛІЗАЦІЇ СИСТЕМИ ТИХОГО РОЗУМНОГО БУДИНКУ

3.1 Дверний модуль

Отже, маючи на руках годящу апаратну платформу для реалізації проєкту, що відповідає висунутим планам та вимогам, можна переходити до реалізації програмної частини проєкту. Та, перш ніж перейти до самої програмної реалізації, іще раз розглянемо загальну концепцію системи, а також місце кожного агрегата, обраного для апаратної реалізації, у цій системі, його призначення та функції.

Пристроєм, що із ним, безсумнівно, найчастіше контактує користувач у даній системі є RFID ридер безконтактних міток. Це невеликий за розмірами пристрій (див. рис. 2.16), що має, за початковим задумом, монтуватися під ручки дверей і зчитувати кільце-мітку, що розташоване на руці користувача, при відкриванні дверей, після чого інформація, зчитана із мітки відправлятиметься на хаб, де оброблюватиметься і запускатиме певні запрограмовані сценарії роботи. Така концепція дозволяє не обтяжувати користувача додатковими маніпуляціями для запуску сценаріїв та подій розумного будинку, а прив'язати їх до виконуваних рутинних користувацьких дій, тобто - відчинення дверей у будинку.

Єдине, що зміниться для користувача у такому випадку - наявність RFID мітки-кільця на руці, що він нею виконує маніпуляції із дверима. Таким чином до роботи із системою можуть бути долучені люди із обмеженими можливостями, як-то порушення мовлення та слуху, адже ці їх обмеження не впливатимуть на використання системи, бо органи із порушеннями не приймають участі у роботі із системою, на відміну від систем із голосовим керуванням. Також системою зможуть користуватися люди, що не бажають спілкуватися із системою вербально в силу певних вад чи вподобань суспільних, релігійних і тд.

					ІАЛЦ.467200.003 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Проте, під час написання даної роботи було прийнято рішення розширити можливі місця для використання ридера RFID і не обмежуватися лише установкою їх під дверні ручки. Таким чином вони можуть бути інстальовані у спеціальні місця у стінах і для активації їх варто лише піднести до цього місця долоню, або ж іншу кінцівку із міткою. Це може бути корисно у ситуаціях від банальних (брудні руки після певних робіт), і до специфічних (відсутність у будинку дверей через вади мобільності чи інші вади користувача). Також дані модулі можуть бути інстальовані у підлогу, а безконтактні мітки - у найнижчі елементи інвалідного візка, через що активація подій відбуватиметься під час проїзду візка із клієнтом повз ридер. Таким чином значною мірою збільшується інклюзивність системи порівняно із початковим задумом, що, звісно ж, не робить її специфічною тільки для користувачів із обмеженнями.

Отже, маємо RFID ридер, що сканує кільце-мітку на руці користувача при відкритті ним дверей. У цей момент постає питання, що і як передаватиме набір нулів та одиниць, що був отриманий в результаті зчитування кільця-мітки на хаб для подальшої обробки.

Для цього застосовується мікроконтролер на базі ESP8266, що був підібраний за низкою критеріїв, описаних у другому розділі даної роботи. Цей мікроконтролер напряду взаємодіятиме із ридером, ініціюватиме зчитування кільця-мітки, отримуватиме від нього, та перетворюватиме у більш зручний для подальших операцій вигляд, номер мітки, й передаватиме його, а також власний ідентифікатор, до хабу для подальшої обробки. Також даний модуль має вбудований світлодіод, що використовуватиметься системою для відображення користувачу стану системи та конкретно цього вузла. Мікроконтролер бере на себе функції встановлення з'єднання, отримання інформації від ридера, її відправки, а також поновлення з'єднання у випадках збоїв системи із різноманітних причин та інформування

					ІАЛЦ.467200.003 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача про позаштатні ситуації у системі.

Так, на даному етапі ми маємо користувача, що відчиняє двері. Під час цього його кільце-мітка знаходиться біля RFID ридера, який зчитує інформацію з нього, і передає по інтерфейсу SPI дані до мікроконтролера ESP8266. Разом RFID ридер та ESP8266 складають “дверний модуль” що інсталується під дверну ручку або ж і в інші місця будинку, як було сказано вище. Він, в свою чергу, є приєднаним до Wi-Fi мережі і при отриманні даних приводить їх до ладу і передає на Raspberry Pi Zero W, що виступає в ролі хабу у даній системі.

Raspberry Pi Zero W - це серце системи “тихого розумного будинку”, тобто її хаб. На цьому мікрокомп’ютері лежить обробка інформації, отриманої від “дверних модулів” та активація відповідних для неї сценаріїв і подій, що своїм результатом має роботу із іншими модулями, актуаторами, а може навіть і сервісами. Наступний рисунок наглядно ілюструє зв’язок між елементами системи.



Рис. 3.1 – Ілюстрація взаємозв’язку елементів системи

Для передачі даних між дверним модулем, тобто ESP8266, до хабу на базі Raspberry Pi Zero W було вирішено використовувати протокол TCP через надійність доправлення даних від модуля-відправника до хабу-отримувача.

Необхідність у цілісності даних на даній ланці зв'язку є крайньою мірою значущою, оскільки від інформації про кільце-мітку (тобто записаного на нього номера) напряму залежить виконання наступних подій саме власника мітки, а не якихось інших, (щодо “мультиюзерності” системи йтиметься у роботі далі), та і виконання у цілому, тому дуже важливою річчю є відсутність спотворення цієї інформації при пересиланні від дверного модуля до хабу, для коректної роботи системи. Завдячуючи використанню протоколу TCP можна бути спокійним за дані, і, в разі виникнення нештатних ситуацій у роботі, шукати та усувати неполадки у роботі або ж самого хабу, або ж при зчитуванні міток. Надалі усунення проблем при зчитуванні міток стало досить актуальною проблемою, тож обрання протоколу передачі даних дало чітко усвідомити місце проблеми і допомогло оперативно та якісно вирішити її.

Для пересилки даних у системі використовувалися сокети. Хаб створює сокети що “слухають” (серверні сокети), і є прив'язаними до певного порту операційної системи, а дверний модуль, в свою чергу, при зчитуванні кільця-мітки користувача, піднімає сокет користувача зі своєї сторони, під'єднується до сокета хабу та передає інформацію, після чого сокет зачиняється до наступного зчитування RFID модулем мітки і появи необхідності у передачі даних у хаб (Рис. 3.2).

На початку розробки були плани тримати зв'язок між “дверним модулем” та хабом постійно відкритим і не закривати сокет після відправки інформації, проте така конфігурація швидко довела свою непрацездатність через недостатній для її нормальної роботи обсяг пам'яті, тому, шляхом оптимізації було вирішено відкривати сокети тільки за необхідності, що є більш правильним способом їх використання.

					ІАЛЦ.467200.003 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

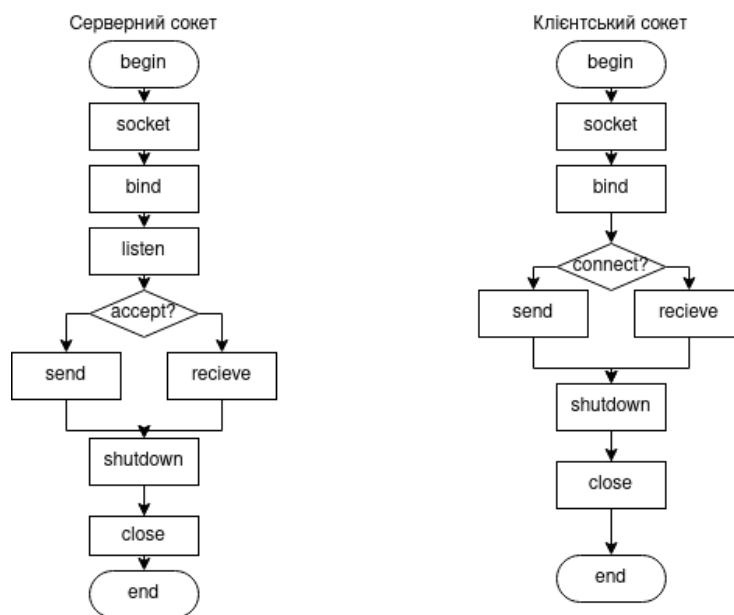


Рис. 3.2 – Схема роботи сокетів: серверного та клієнтського

Для реалізації програмної складової на ESP8266 у “дверному модулі” було обрано мову програмування MicroPython. MicroPython - це ошадлива та ефективна реалізація мови програмування Python 3, оптимізована для роботи на мікроконтролерах, яка включає деяку підмножину стандартних бібліотек Python. Обрано цю мову було виключно через її перспективність, адже, не дивлячись на свою достатню новизну, ця реалізація усім відомої, шанованої та мегапопулярної мови програмування Python у вкрай малий термін привернула до себе увагу багатьох розробників і наразі дуже активно ними розвивається, чітко окреслюючи сферу свого застосування, та відбираючи у мастодонтів серед мов для мікроконтролерів усе більшу долю ринку із кожною новою версією. При своїй, традиційно пайтоновській, зручності для розробника, MicroPython майже не поступається своїм більш досвідченим сусідам по сфері застосування. Отже, саме MicroPython було обрано для програмування цього вузла системи, а його функціоналу виявилось більш ніж достатньо. Також значним плюсом при програмуванні даного мікроконтролера за допомогою MicroPython виявилася наявність класичного пайтонівського командного рядка інтерпретатора, що значно допомагає при

розробці та тестуванні, адже, за допомогою нього можна слідкувати за поведінкою програми у рантаймі, відстежувати, як обробляються дані та які дані на мікроконтролер передаються від RFID модуля.

Також командний рядок інтерпретатора дає можливість тестувати виконання окремих рядків програмного коду, що нерідко допомагає зрозуміти вади форматування даних, або ж обрати із різних варіантів реалізації найкращий. Також за його допомогою можна отримати базові корисні відомості про мікроконтролер та його стан, підключення до мережі, його IP-адресу, зайнятість пам'яті контролера і ще багато чого, не кажучи вже про можливість м'якого перезапуску мікроконтролера, не вдаючись до використання піну reset чи відключення живлення, що у ситуаціях розробки програми для мікроконтролера робить її суттєво зручнішою та економить час, адже розробнику не потрібно відриватися від комп'ютера, за допомогою якого здійснюється програмування мікроконтролера, аби перезапустити його при виникненні певних критичних помилок, як, скажімо, брак вбудованої пам'яті через неакуратне поводження із нею розробника, або ж неочікувану роботу програмного коду.

```
MicroPython v1.15 on 2021-04-18; ESP module (1M) with ESP8266
Type "help()" for more information.
>>> help()

Welcome to MicroPython!

For online docs please visit http://docs.micropython.org/en/latest/esp8266/ .
For diagnostic information to include in bug reports execute 'import port_diag'.

Basic WiFi configuration:

import network
sta_if = network.WLAN(network.STA_IF); sta_if.active(True)
sta_if.scan() # Scan for available access points
sta_if.connect("<AP_name>", "<password>") # Connect to an AP
sta_if.isconnected() # Check for successful connection
# Change name/password of ESP8266's AP:
ap_if = network.WLAN(network.AP_IF)
ap_if.config(essid="<AP_NAME>", authmode=network.AUTH_WPA_WPA2_PSK, password="<password>")

Control commands:
CTRL-A -- on a blank line, enter raw REPL mode
CTRL-B -- on a blank line, enter normal REPL mode
CTRL-C -- interrupt a running program
CTRL-D -- on a blank line, do a soft reset of the board
CTRL-E -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)

>>> |
```

Рис. 3.3 – Командний рядок інтерпретатора MicroPython із даними про мікроконтролер у ньому

Отже на ESP8266 був завантажений відповідний firmware (вбудована програма, “прошивка”) та мікроконтролер був підключений до комп’ютера за допомогою перехідника USB-UART для подальшої із ним роботи.

При написанні програмного забезпечення для мікроконтролера ESP8266 використовувалася бібліотека роботи із ридером RFID міток MFRC522 оскільки обраний у другому розділі ридер безконтактних міток заснований на мікросхемі MFRC522. Дана бібліотека ініціалізує роботу із RFID ридером за протоколом SPI та дозволяє отримувати інформацію від нього через задані піни. В наступній таблиці позначено відповідність GPIO пінів мікроконтролера сигнальним пінам ридера.

Таблиця 3.1 – Комутація виходів RFID ридера із пінами ESP8266

Сигнальні піни ридера	GPIO піни ESP8266
sck	0
mosi	2
miso	4
sda	14
rst	5

З’єднавши мікроконтролер із ридером, згідно таблиці, за допомогою проводів, ми даємо змогу мікроконтролеру комунікувати із ридером. Для отримання від ридера інформації про вміст мітки (кільця, картки чи брелока) використовується модуль read.py, функція do_read() якого ініціалізує підключення ридеру до мікроконтролера, активує піни підключення контролера до ридеру для передачі даних та зчитування мітки. Після чого чекає, доки мітка не буде зчитана за допомогою функції read() із бібліотеки і діє по-різному у 3-х випадках:

1. Якщо картка була зчитана і при цьому процесі не виникло проблем, то функція do_read() повертає строку, що містить ідентифікатор модуля у форматі ‘door<№>’ для того, аби надалі

2. система мала змогу зрозуміти, від якого саме модуля була отримана інформація і в залежності від цього виконати необхідні дії, а також містить індивідуальний номер мітки, що складається зі 16 чисел у форматі списку. В цілому, строка, що повертається, має наступний вигляд: 'door<№>: [<№ картки>]' (наприклад: 'door1: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]')
3. Якщо мітка, що була зчитана ридером, мала інший формат номеру, відмінний від необхідного, тобто мітка не має відношення до системи, то функція повертає повідомлення про помилку аутентифікації.
4. У випадку ж, коли відбулася помилка при самому зчитуванні мітки, функцією повертається повідомлення про те, що не вдалося обрати тег.

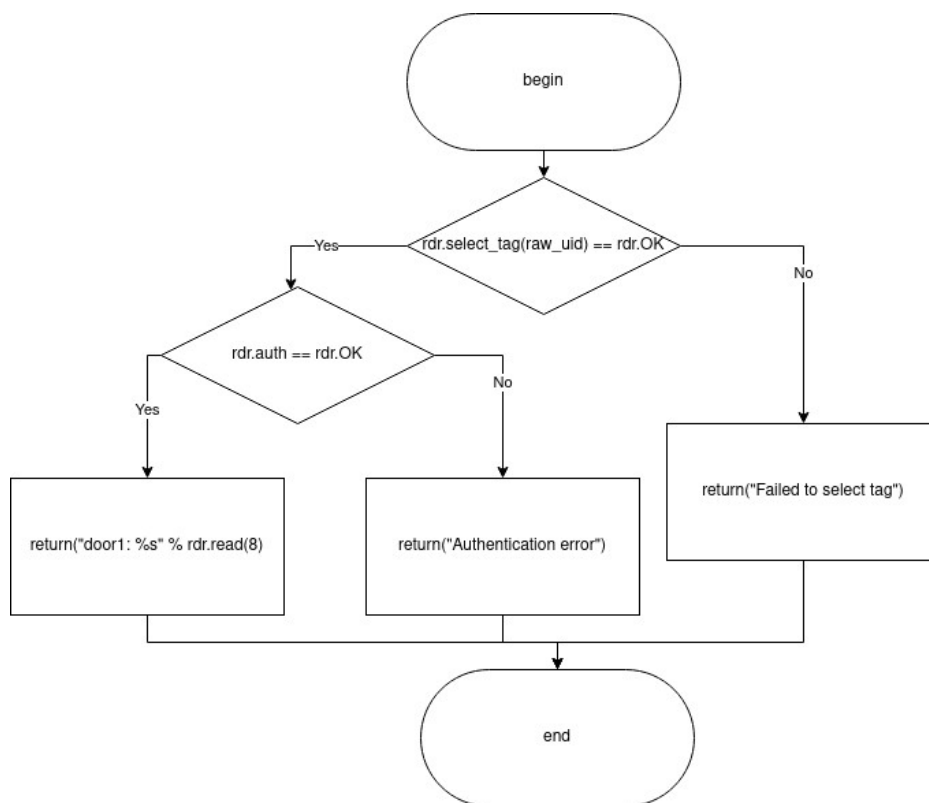


Рис. 3.4 – Блок-схема алгоритму роботи модуля зчитування міток

Змн.	Арк.	№ докум.	Підпис	Дата

Також для тестування та подальшої роботи системи при розробці використовується модуль `write.py`, який, як слідує із назви, записує на мітки задані розробником номери. При подальшій роботі системи даний модуль не є необхідним, оскільки не передбачається, що користувач матиме можливість створювати додаткові мітки, тому після закінчення цього етапу розробки модуль `write.py` видаляється зі внутрішньої пам'яті мікроконтролера задля оптимізації.

Модуль `write.py` дещо схожий до свого зчитувального побратима. Його головна функція `do_write()` також ініціалізує роботу із ридером та його підключення до відповідних пінів і повторює усі інші дії аж до самого моменту запису інформації на картку, де, замість функції `read()` із відповідної бібліотеки викликається функція `write()`, що передає на запис строку у байтовому форматі із 16 чисел і також має декілька вірогідних варіантів роботи, що представлені на рисунку (Рис. 3.5).

Даний модуль запису інформації на мітки побудований таким чином, щоб його було зручно використовувати у командному рядку інтерпретатора `MicroPython`. Достатньо ввести необхідний номер безконтактної мітки прямо у відповідне місце файлу `write.py` відповідним чином (наприклад, для того, аби записати на мітку номер [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] потрібно у другий аргумент функції `write()` передати значення `b"\x0f\x0e\x0d\x0c\x0b\x0a\x09\x08\x07\x06\x05\x04\x03\x02\x01\x00"`). Після чого викликати виконання функції у командному рядку інтерпретатора, піднести до RFID ридера картку, кільце чи інший різновид мітки та дочекатися повідомлення про вдалий запис у вікні командного рядка.

					ІАЛЦ.467200.003 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

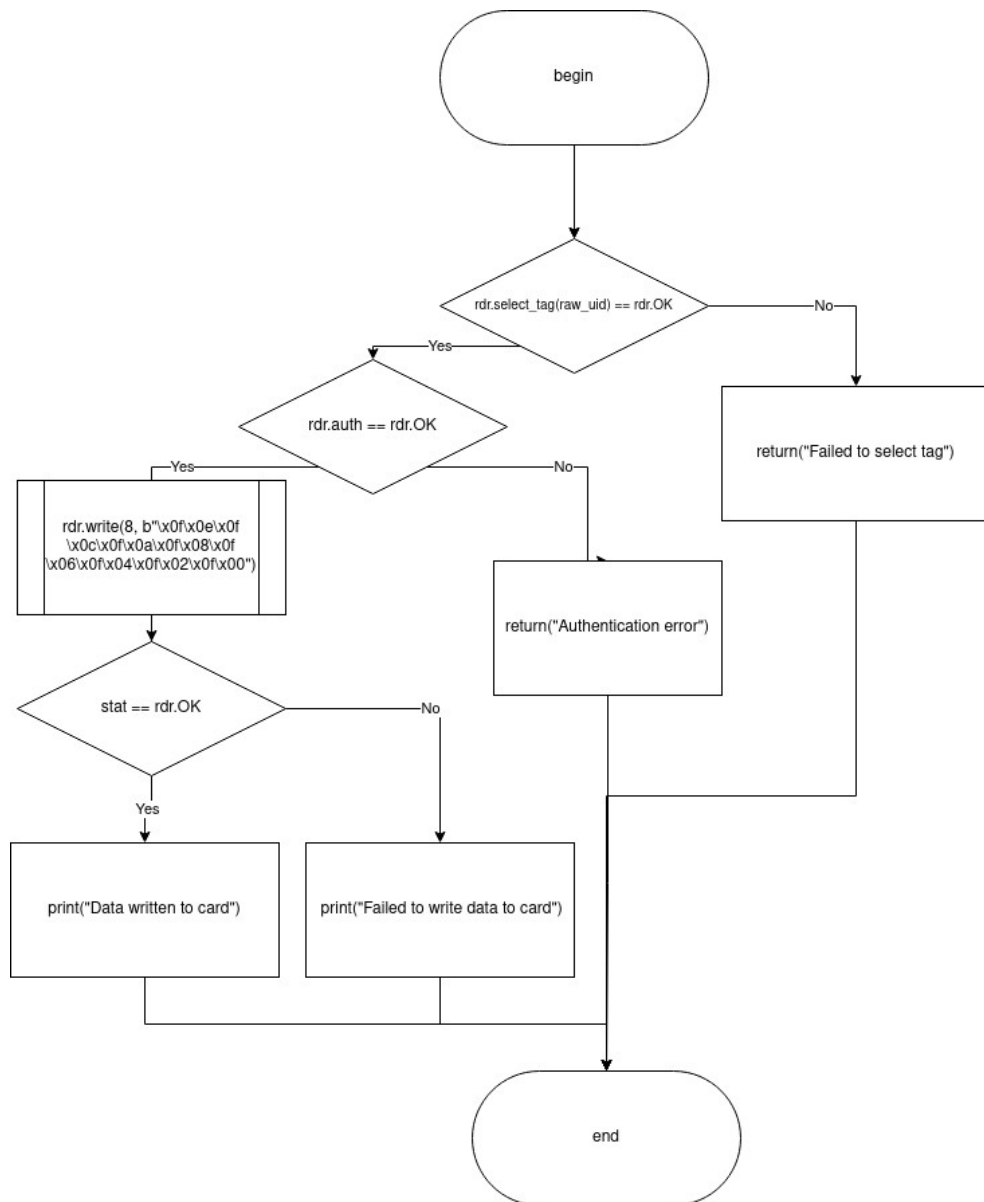


Рис 3.5 – Блок-схема алгоритму роботи модуля запису безконтактних міток

При виникненні помилки варто піднести мітку ще раз, або замінити її, якщо в цьому є необхідність. На наступному рисунку (Рис. 3.6) зображений процес запису на порожню мітку номера [15, 14, 15, 12, 15, 10, 15, 8, 15, 6, 15, 4, 15, 2, 15, 0] та продемонстрована дієвість цієї операції за допомогою подальшого зчитування номера, із якого видно, що зчитаний номер картки співпадає із записуваним.

```

MicroPython v1.15 on 2021-04-18; ESP module (1M) with ESP8266
Type "help()" for more information.
>>> import write
>>> write.do_write()

Warning: SPI(-1, ...) is deprecated, use SoftSPI(...) instead

Place card before reader to write address 0x08

Data written to card
Bye

>>> import read
>>> read.do_read()

Warning: SPI(-1, ...) is deprecated, use SoftSPI(...) instead
'door1: [15, 14, 15, 12, 15, 10, 15, 8, 15, 6, 15, 4, 15, 2, 15, 0]'

```

Рис. 3.6 – Приклад запису інформації на безконтактну мітку

Отже, ми розглянули інструменти, за допомогою яких мікроконтролер співпрацює із RFID ридером і отримує/записує інформацію із безконтактних міток. Тепер перейдемо до основної частини програмного забезпечення.

Основою програмного забезпечення мікроконтролера дверний модуль ESP8266 є модуль transfer.py, що дістав своє ім'я через своє основне призначення у даному програмному забезпеченні, а саме - пересилку отриманих від RFID ридера даних, проте його функції включають також інші дії у мережі, а також обробку помилок.

Алгоритм роботи модуля представлений на наступному рисунку (Рис. 3.7):

					ІАЛЦ.467200.003 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

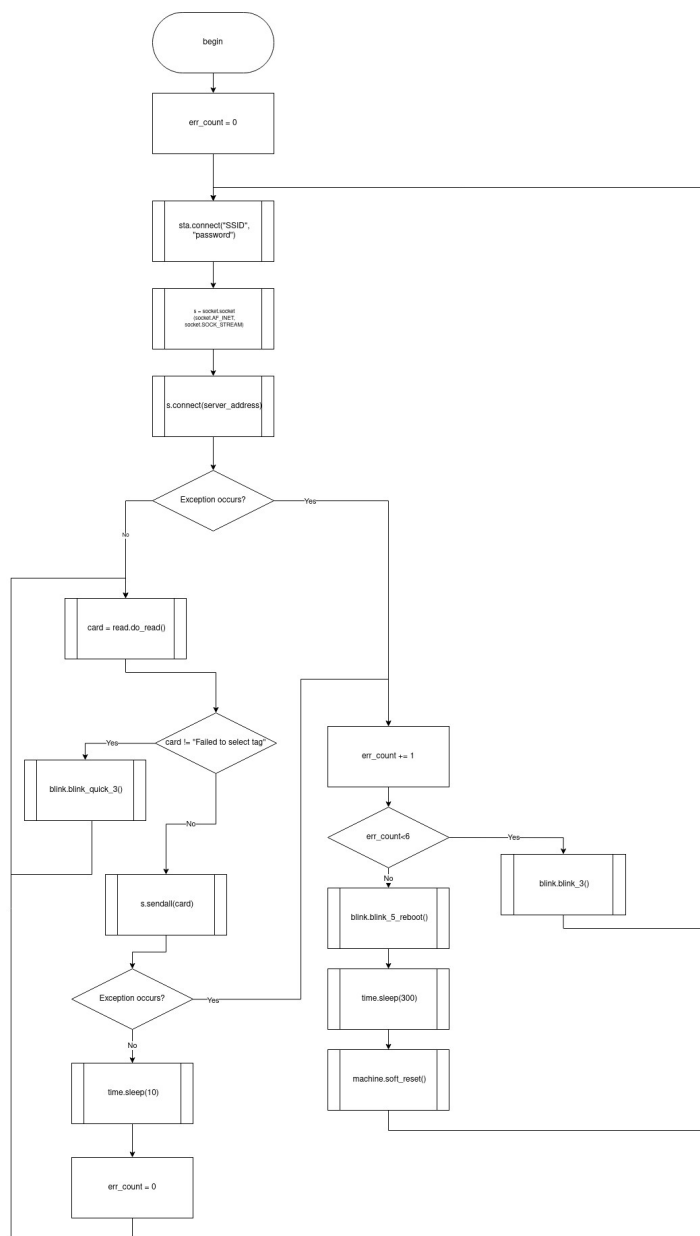


Рис. 3.7 – Блок-схема алгоритму роботи модуля transfer.py

В таблиці 3.2 наведено перелік використовуваних під час роботи даного модуля функцій, та їх завдань і наслідків роботи.

Таблиця 3.2 – Функції модуля transfer.py

Функція	Використання
sta.connect('SSID', 'password')	Використовується для підключення мікроконтролера до бездротової мережі із заданими SSID та password
s.connect(server_address)	Підключає сокет до вказаної у server_address адреси

Продовження таблиці 3.2

Функція	Використання
s.sendall(card)	Відсилає за протоколом TCP дані зі змінної card, тобто строку, яка містить ідентифікатор дверний модуль (номер дверей) та номер безконтактної мітки, що її було зчитано, згідно із описаним вище алгоритмом, використовуючи ініціалізований раніше сокет s
time.sleep()	Призупиняє роботу модуля на задану в дужках кількість секунд
blink.blink_3	Тричі довго блимає світлодіодом, сповіщаючи користувача, що виникла проблема зі з'єднанням, після чого чекає деякий час і повертається назад в основне тіло програми, для того, аби іще раз спробувати підключитися до хабу. Такий алгоритм поведінки був виведений дослідним шляхом а час очікування підібраний таким чином, аби програма на хабі, у випадку свого "падіння" мала час перезапуститися і знову увімкнути слухаючий сокет на потрібному порту.
blink.blink_quick_3()	Змушує світлодіод на контролері швидко блимнути тричі, сповіщаючи користувача про те, що зчитування мітки не відбулося коректно і даючи йому можливість перезчитати мітку.
blink.blink_5_reboot()	Якщо після 5-ти спроб очікування та перепідключення до хабу відновити зв'язок так і не вдалося попри усі намагання системи, програма викликає дану функцію, яка відповідними короткими (по пів секунди) п'ятьма блимами світлодіода дверного модуля сповіщає користувача про те, що усунути цю проблему самотужки системі не вдалося і потрібне його втручання у систему для перезапуску хабу, після чого дверний модуль впадає у режим очікування на 5-ть хвилин. За цей час користувач має перевірити хаб, його живлення, а також стан бездротової мережі і прийняти певні рішення та дії для усунення знайдених проблем. Після 5-ти хвилинного очікування програма викликає системну функцію із бібліотеки machine - soft_reset() і перезапускає модуль, вважаючи, що користувач усунув проблему і можна знову починати роботу, після чого робота програмного забезпечення на ньому починається з початку. Таким чином система досягає автономності у вирішення більшості проблем зі зв'язком, що виникають під час її роботи, і потребує втручання користувача лише у крайніх випадках.
machine.soft_reset()	Виконує м'який зброс мікроконтролера, видаляючи усі Python-об'єкти

Кінець таблиці 3.2

Останніми не розглянутим елементом програмного забезпечення дверного модуля є файл main.py та boot.py. Ці файли виконуються кожного разу після запуску системи, тобто після подачі живлення. Із початку, як тільки з'явилося живлення, MicroPython на мікроконтролері ESP8266 починає свою роботу, активує необхідні модулі та процеси і виконує інші,

необхідні для подальшої роботи речі, виконання яких частково прописане у файлі `boot.py`, зміни у який не вносилися. Після вдалого виконання `boot.py` MicroPython загрузжений і готовий до роботи на мікроконтролері, і автоматично виконує `main.py`. Розробники MicroPython рекомендують писати програмний код у цьому файлі лише при виконанні зовсім невеликих проєктів, а у інших випадках просто ініціалізувати у цьому модулі виконання інших. Тому, згідно із рекомендаціями, програма у даному файлі лише 10 разів швидко блимає світлодіодом, сповіщаючи користувача про те, що MicroPython нормально завантажився на мікроконтролері і починає роботу, після чого одразу ж переходить до виконання основного коду дверного модуля, розташованого у `transfer.py`.

Порядок роботи модулів та їх взаємозв'язок показано на наступному рисунку (Рис 3.8):

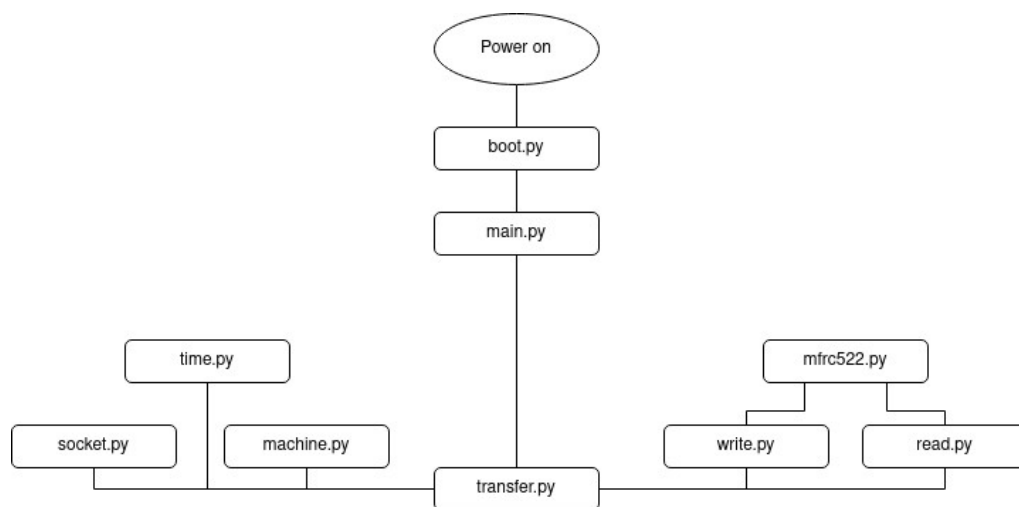


Рис. 3.8 - Діаграма залежностей модулів

Отже, загалом, мікроконтролер дверного модуля запрограмований на наступну роботу:

1. Після підключення живлення проходять стандартні процеси запуску MicroPython, після чого мікроконтролер світлодіодом сповіщає користувача про вдале завантаження та початок роботи.

2. Далі мікроконтролер підключається до бездротової мережі, ініціалізує сокет так підключається із його допомогою до хабу.
3. Далі мікроконтролер ініціалізує зчитування безконтактної мітки RFID ридером і очікує, коли користувач піднесе її до робочої зони ридера, і той проведе зчитування номеру.
4. Якщо зчитати номер мітки не вдалося модуль іще раз намагається виконати цю операцію.
5. Коли зчитування мітки пройшло успішно програма мікроконтролера формує отриманий номер мітки у строку, додає туди власний ідентифікатор модуля та відсилає цю інформацію на хаб за протоколом TCP. Після цього повертається назад до 3 пункту.
6. Якщо підключитися до сокета, або мережі, або відправити інформацію не вдалося, тоді мікроконтролер сповіщає користувача про збій у роботі і 5 разів намагається самотужки відновити з'єднання із мережею/хабом, після чого, в разі удачі цих спроб повертається до пункту 3.
7. Якщо після 5-ти спроб мікроконтролеру так і не вдалося поновити з'єднання, тоді він сповіщає користувача про необхідність його втручання у роботу системи і входить у режим очікування на 5-ть хвилин.
8. Після виходу із режиму очікування програмне забезпечення мікроконтролера виконує ребут, повертаючись до 1 пункту.

3.2 Хаб

Тож дверний модуль виконав свою роботу, успішно зчитав кільце-мітку на руці користувача (або ж не успішно і просигналізував про це, але припустимо, що результат усе ж таки призвів до виконання запланованого

					ІАЛЦ.467200.003 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

рамках концепції IFTTT після чого відправляти керуючі сигнали, що відповідають певним діям у частині “That” даної концепції.

Отже, спершу хаб має визначити першу змінну у рамках IFTTT концепції - частину “This”, що він її отримує від дверного модуля і на основі неї, власне, і переходить до виконання частини “That”. А для того, щоб її визначити треба, перш за все, її отримати.

Для отримання інформації від дверних модулів використовується бездротовий зв’язок за стандартом Wi-Fi.

При розробці було передбачено два можливих варіанти з’єднання між собою дверних модулів, хабу та кінцевих виконавчих пристроїв - актуаторів:

- Хаб виступає точкою доступу самостійно і запускає власну бездротову мережу із SSID “SilentSmart”, до якої підключаються всі дверні модулі та актуатори і здійснюють обмін даними у межах цієї мережі. Обмін даними в окремій мережі звісно є значним плюсом із багатьох різних факторів, у тому числі й безпеки даних, проте Raspberry Pi Zero W має лише один бездротовий Wi-Fi інтерфейс, а тому такий варіант використання виключає доступ системи тихого розумного будинку до інтернету, що може бути потрібним при використанні різноманітних онлайн-сервісів чи інтеграції системи тихого розумного будинку у іншу, скажімо, при роботі із Google Nest, де усі керуючі сигнали відправляються не напряму до актуаторів, а через сервери Google POST-запитами за допомогою Nest API. Саме для випадків необхідності інтернет-підключення і передбачена друга варіація роботи.
- Хаб разом із дверними модулями та актуаторами використовують бездротову мережу, запущену стороннім пристроєм (найчастіше - простий домашній Wi-Fi). Це, звісно, поміщає інформацію, що

					ІАЛЦ.467200.003 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

- курсує у системі у спільну із іншими даними мережу, проте дає хабу доступ до мережі Інтернет, що є суттєвим плюсом, адже тоді він може мати набагато ширші можливості, і не обмежуватися кінцевими подіями виключно у вигляді команд актуаторам.

Отже користувач, в залежності від своїх потреб, може власноруч обирати режим, в якому працюватиме система. Перший - коли йому потрібно лише керувати безпосередньо доступними домашніми актуаторами розумного будинку, і другий, коли потрібна взаємодія через Інтернет для контролю певних пристроїв, чи виконання певних подій (скажімо, відправка певних повідомлень у месенджері тощо).

Так ми маємо певну бездротову мережу (запущену або самим хабом, або третім пристроєм), до якої приєднаними є хаб та дверний модуль. Як вже зазначалося раніше у підрозділі про дверний модуль, хаб має сокет, що слухає на певному порті і очікує підключення від дверного модуля, аби прийняти інформацію.

При спрацюванні дверного модуля він відкриває зі своєї сторони сокет і встановлює з'єднання із відповідним портом хабу (див. Рис. 3.2). Для роботи сокета було обрано 4006 порт хабу, інформація про який зберігається у змінній port задля того, аби порт можна було зручно змінити за необхідності. Далі за протоколом TCP до хабу передається інформація, що містить ідентифікатор дверного модуля та номер кільця-мітки користувача. Після вдалого отримання цієї інформації сокет закривається і далі отримана інформація переходить у обробку. Детальна блок-схема роботи програмного забезпечення хабу представлена у Додатку).

Проте під час розробки виникли певні труднощі із роботою слухаючого сокета, оскільки при простому його використанні після отримання інформації від дверного модуля і закриття сокета подальше підключення до нього було проблематичним, до того ж якщо даний сокет був

зайнятий, то хаб не міг прийняти інформацію від іншого модуля, що створювало значні незручності при використанні системи більш ніж одним користувачем (а подібний сценарій використання також передбачався), оскільки тепер декілька користувачів не мали змоги відчиняти двері одночасно. Тому виникла необхідність побудувати роботу таким чином, аби декілька користувачів могли відчиняти двері одночасно, а хаб - одночасно приймати інформацію від дверних модулів.

Одразу ж був вигаданий та розглянутий варіант, коли хаб тримає лише один сокет слухаючим і при підключенні клієнта (тобто - дверного модуля) за цим сокетом повідомляє йому адресу іншого порту, на якому відкриває сокет спеціально для роботи із цим клієнтом, а даний сокет закриває і починає чекати на інформацію там.

Проте дана концепція майже одразу була облишена, оскільки видавалась не досить надійною, а також не вирішувала усіх наявних проблем. До того ж реалізація клієнтської частини на мікроконтролері ESP8266 у дверному модулі виявлялася дещо обтяжливою для нього, що є безперечним мінусом.

Також був розглянутий варіант із використанням паралелізму і запуском сокетів у різних потоках, проте врешті було вирішено реалізовувати ідею мільтисервера на сокетах у неблокованому режимі та event loop (Рис. 3.10).

В ньому слухаються декілька сокетів і коли на один із них встановлюється зв'язок і приходить інформація, то викликається колбек, який передає цю інформацію далі у роботу, а сокети продовжують слухати надалі. Після передачі колбеком інформації, що була отримана сокетом від дверного модуля починається її обробка. Детальна схема роботи програмного забезпечення хабу наведена у додатках.

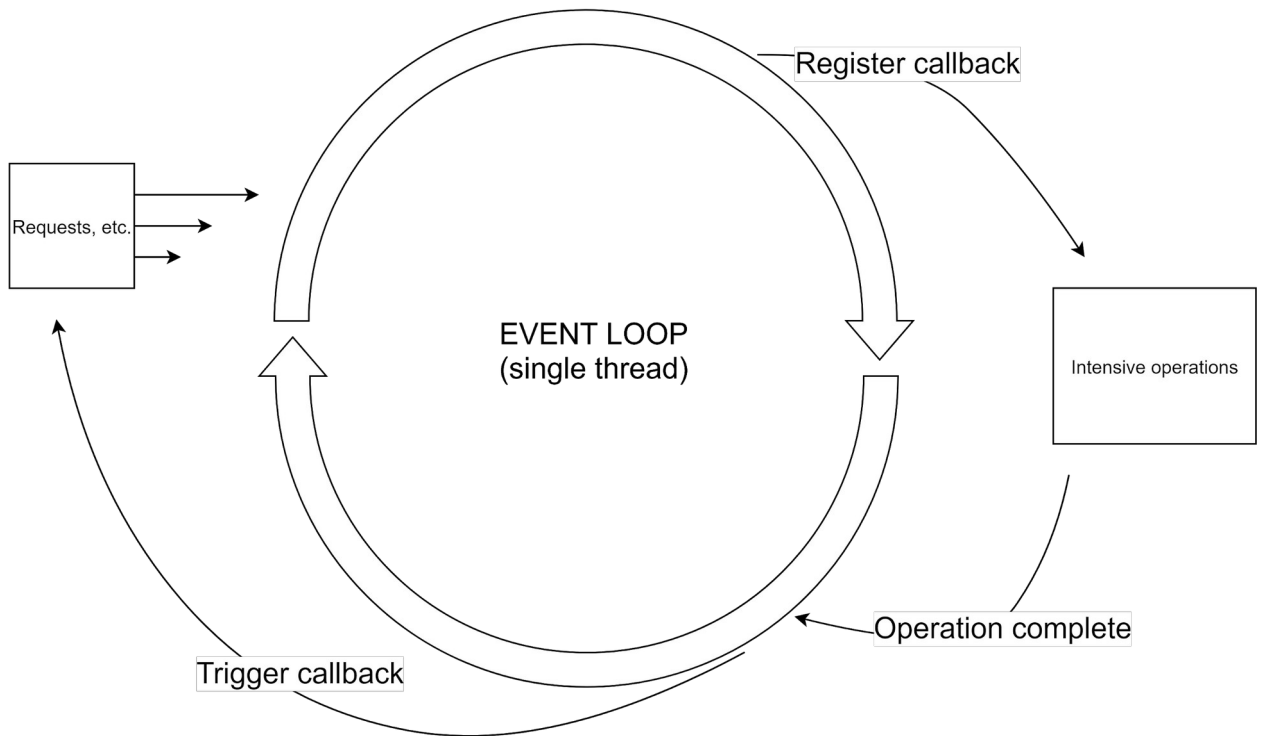


Рис. 3.10 – Візуальне представлення роботи event loop

Пояснення деяких важливих елементів під час роботи програмного забезпечення хабу наведені у таблиці 3.3:

Таблиця 3.3 – призначення частини коду програмного забезпечення хабу

Функція, умовний перехід або дія	Призначення
<code>if gotten_data is not None</code>	Іноді хаб може отримувати від дверних модулів зовсім порожні пакети інформації, що їх необхідно відсіювати, тому перш за все перевіряється наявність даних в отриманому повідомленні узагалі
<code>if gotten_data != b'None'</code>	Іноді при помилках зчитування безконтактних міток дверний модуль до хабу може відправити інформацію у вигляді <code>b'None'</code> , або <code>b'door<№>: None'</code> . Для того, аби не пропускати подібні дані у роботу відбувається їх перевірка за допомогою цього умовного переходу, що забезпечує від потрапляння даних першого роду у обробку
<code>gotten_data = gotten_data[:gotten_data.index('')+1]</code>	Якщо користувач із якихось причин не прибере від дверного модуля безконтактну мітку, то до хабу може передана інформація із повтором, що виглядає наступним чином: <code>"b'door1: [15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]door1: [15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]"</code> , тому при роботі хабу використовується обмеження даних по перший символ "]", що забезпечить хаб від потрапляння в обробку "подвійних доз" інформації.

Продовження таблиці 3.3

Функція, умовний перехід або дія	Призначення
<code>username = rules.users[str(card_id)]</code>	Отримує зі словника <code>rules.users[]</code> ім'я користувача, відповідно до отриманого номеру мітки
<code>actions = rules.rules_dict[username]</code>	Отримує зі словника <code>rules.rules_dict[]</code> ймовірні варіанти кінцевих функцій для даного користувача
<code>actions[door_id]()</code>	Виконує функцію відповідну для даного користувача при активації даного дверного модуля
<code>actions.no_user()</code>	Функція, що спрацьовує у випадку використання користувачем у системі невідомої або порожньої безконтактної мітки. Розроблена таким чином, що може бути змінена за бажанням користувача, і поводити себе по-іншому, наприклад, надсилати користувачу повідомлення про подібний випадок, або призводити до блокування всієї системи у разі несанкціонованого втручання із боку зловмисників.
<code>ConnectionRefusedError</code>	Виникає у разі відсутності зв'язку із кінцевим пристроєм-виконавцем (актуатором) і викликає повідомлення користувача про те, що актуатор є недоступним і його потрібно перевірити

Кінець таблиці 3.3

Для роботи програми хабу ключовими є модулі `rules` та `actions`:

- Модуль `rules` зберігає у собі два основоположних словники цієї програми - `users` та `rules_dict`:
 - Словник `users` зберігає у собі відповідності адрес міток до імен користувачів задля забезпечення простоти коду і подальшої можливості оперувати набагато більш зручними іменами користувачів, замість довгих і громіздких списків із номером безконтактної мітки.
 - Словник словників `rules_dict` зберігає у собі словник відповідностей різноманітних сценаріїв у системі до імен користувачів. Правила у системі є також словником і складаються із ідентифікатора дверного модуля та посилання на дію із модуля `actions`. Тобто загалом даний словник зберігає відповідність імен до ідентифікаторів дверних модулів та подій із модуля `actions`.

- Модуль actions, в свою чергу, зберігає у собі кінцеві функції, доступні користувачам, тобто ту саму частину “That” із концепції IFTTT. Це функції що відправляють команди актуаторам або виконують інші необхідні користувачу дії.

3.3 Модулі тестування

Також додатково було розроблено три модулі для тестування роботи системи: client_imitation, multicient та actuatur_imitation. Їх склад та призначення детально розписані в наступній таблиці 3.4:

Таблиця 3.4 – Призначення модулів тестування

Модуль	Використовувані модулі	Призначення
client_imitation	socket, time	Імітує роботу дверного модуля у тому вигляді, як її сприймає хаб системи. Даний модуль створює сокет та підключається до хабу, видаючи себе дверним модулем, відправляє на хаб дані у форматі відповідні тому, у якому їх відправляє справжній віддалений модуль, що працює на ESP8266 у парі з RFID ридером, після чого завершує свою роботу. Модуль був створений для того аби імітувати поодинокі відкривання дверей користувачем і відтестувувати поведження системи при різноманітних вхідних даних від дверних модулів, у тому числі й помилкових.
multicient	socket, selectors, time, types	Модуль, аналогічний за ідеєю до модуля client_imitation, проте імітує роботу не одного поодинокого модуля, а декількох дверних модулів одночасно або через невеликі проміжки часу. Даний модуль був створений для тестування програмного забезпечення хабу в умовах підключення багатьох дверних модулів, а також для тестування сталості його роботи.
actuatur_imitation	socket, time	Імітує роботу кінцевого пристрою даної системи - актуатора, що є кінцевою ланкою роботи системи і безпосередньо його робота споглядається користувачем. Саме актуатор вимикає світло, чи відправляє повідомлення тощо. Модуль actuatur_imitation був створений із метою перевірки роботи програмного забезпечення хабу із зовнішніми пристроями та роботу подій із модуля actions, і представляє собою просту програму, що має слухаючий сокет, що приймає від хабу керуючі сигнали через мережу і, в залежності від них, виводить в консоль різноманітні повідомлення.

Використання модуля `actuatur_imitation` призвело до виявлення та усунення несправності що описана в таблиці 3.3 у графі “`ConnectionRefusedError`”: коли при спробах виконати відповідну подію, що відправляє на актуатор керуючий сигнал, виявлялося, що актуатор не слухає у даний момент, і взагалі недоступний, то робота програмного забезпечення хабу повністю припинялася. Після роботи над даною помилкою таких прикостей більше не трапляється, і хаб повідомляє користувача про те, що, очевидно, актуатор є недоступним і його варто перевірити.

					ІАЛЦ.467200.003 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок до розділу 3

Отже, у третьому розділі даної роботи була проведена праця по реалізації дипломного проекту, а також її опис, в ході якої було реалізовано програмно-апаратний комплекс системи тихого розумного будинку. Під час виконання роботи над ним виникло чимало складнощів, що були успішно вирішені. Рішення деяких із них було наведено у цьому розділі даної роботи, де також детально описано саму концепцію, що на момент написання була вже остаточно сформованою, а також алгоритми та способи роботи створеного програмно-апаратного комплексу, разом із завданнями, що мають вирішувати як сама система в цілому, так і її окремі модулі. Усе це підкріплено значною кількістю ілюстративних матеріалів, що мають спростувати читачу сприйняття роботи системи проекту. В цілому висновок із приводу реалізації початкової ідеї є втішним, оскільки програмно-апаратний комплекс системи тихого розумного будинку працює і виконує покладені на нього функції. Не дивлячись на це результат не варто вважати повним, оскільки система тихого розумного будинку має великий потенціал для подальшого доповнення та модернізації, що передбачає розробку повноцінного інтерфейсу користувача, а також додавання до системи інших модулів, у тому числі і модулів із інших систем розумного будинку, наприклад Google Nest. Загалом же, не дивлячись на поки що вузький функціонал, у цьому розділі заклався фундамент системи, що має шалені перспективи для подальшого розвитку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

ВИСНОВОК

У даній роботі було досліджено історичну генезу концепції "розумного дому", а також "інтернету речей" від самих зародкових часів цих концепцій до сьогодні, і на її основі, беручи до уваги, звісно, найуспішніших комерційних представників сучасного ринку "розумних речей" були виокремлені характерні риси та тенденції систем "розумного будинку" і напрямки перспективного розвитку. Одним із найпривабливіших напрямків був варіант, пов'язаний із користувачами з обмеженими можливостями, його в решті і було обрано для реалізації інклюзивної системи "розумного будинку". Проект було вирішено назвати "тихим" через відсутність під час його використання голосових команд, що дає можливість користуватися цією системою людям із порушеннями мовлення та слуху і не тільки.

Після остаточного формування концепції було висунуто вимоги до апаратної реалізації та проведено ґрунтовний аналіз наявних на ринку апаратних складників, придатних для реалізації системи, у ході якого вибудувалася апаратна складова майбутнього програмно-апаратного комплексу проекту.

Після вибору апаратного базису для реалізації проекту "тихого розумного будинку" була проведена робота по розробці та написанню програмного забезпечення для роботи даної системи. Під час тестування та налагодження програмно-апаратного комплексу проекту було усунуто усі наявні проблеми у роботі, а також відкриті нові горизонти для роботи системи: її розширення актуаторами та модулями, інтеграція у інші системи "розумного будинку" від сторонніх розробників тощо.

Усі етапи роботи над проектом були детально задокументовані та описані у пояснювальній записці до даного проекту.

					ІАЛЦ.467200.003 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

За час роботи над проектом вдалося реалізувати початковий замисел, скоригувавши його по ходу дослідження відповідно до вимог сучасності у сфері концепцій "розумного будинку" та "інтернету речей", наслідком чого стала реалізація як апаратної, так і програмної складової проєкту. Програмно-апаратний комплекс створеного проєкту є дієздатним і виконує покладені на нього під час формування концепції та розробки функції.

До того ж, очевидними є перспективи розвитку системи "тихого розумного будинку" у її інклюзивній ніші, тому можна вважати, що дана робота закладає фундамент для майбутніх досліджень та різноманітних апгрейдів системи.

					ІАЛЦ.467200.003 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Arduino - ArduinoBoardBT. *Arduino - Home*. URL: <https://www.arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth> (дата звернення: 27.04.2021).
2. Arduino Official Store. *Arduino nano 33 iot. Arduino Official Store | Boards Shields Kits Accessories*. URL: <https://store.arduino.cc/arduino-nano-33-iot> (дата звернення: 26.04.2021).
3. Arduino Official Store. *ARDUINO UNO WiFi REV2. Arduino Official Store | Boards Shields Kits Accessories*. URL: <https://store.arduino.cc/arduino-uno-wifi-rev2> (дата звернення: 26.04.2021).
4. Arduino Uno Wi-Fi Rev 2: распиновка, схема подключения и программирование. *База знаний Амперки*. URL: <http://wiki.amperka.ru/products:arduino-uno-wifi-rev-2> (дата звернення: 26.04.2021).
5. Buegiga Technologies Oy. Документация на серию wt11-e. *Компэл*. URL: <https://www.compel.ru/item-pdf/5baf4eb9f08be74ad33fb4f961924071/ps/bluegiga~wt11-e.pdf> (дата звернення: 26.04.2021).
6. Configurable ZigBee-based control system for people with multiple disabilities in smart homes / R. Aburukba та ін. *2016 international conference on industrial informatics and computer systems (CIICS)*, м. Sharjah, Dubai, United Arab Emirates, 13–15 берез. 2016 р. 2016. URL: <https://doi.org/10.1109/iccsii.2016.7462435> (дата звернення: 12.04.2021).
7. Ejaz W., Anpalagan A. *Internet of things for smart cities*. Cham : Springer International Publishing, 2019. URL: <https://doi.org/10.1007/978-3-319-95037-2> (дата звернення: 12.04.2021).

					ІАЛЦ.467200.003 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Espressif Inc. ESP8266-DevKitC getting started guide. *Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems*. URL: https://www.espressif.com/sites/default/files/documentation/ESP8266-DevKitC_getting_started_guide_EN.pdf (дата звернення: 27.04.2021).
9. Espressif Inc. Esp8266. *Wayback Machine*. URL: https://web.archive.org/web/20171209024011/http://espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf (дата звернення: 27.04.2021).
10. Espressif Inc. Modules | espressif systems. *Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems*. URL: <https://www.espressif.com/en/products/modules> (дата звернення: 27.04.2021).
11. Gillis A. S. What is iot (internet of things) and how does it work?. *IoT Agenda*. URL: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> (дата звернення: 15.03.2021).
12. Hensel B. K., Demiris G. Technologies for an aging society: a systematic review of “smart home” applications. *Yearbook of medical informatics*. 2008. Т. 17, № 01. URL: <https://doi.org/10.1055/s-0038-1638580> (дата звернення: 12.04.2021).
13. Hill J. The smart home: a glossary guide for the perplexed. *T3*. URL: <https://www.t3.com/features/the-smart-home-guide> (дата звернення: 15.03.2021).
14. ITU. Measuring digital development - mobile vs. fixed. *ITU*. URL: <https://itu.foleon.com/itu/measuring-digital-development/mobile-vs-fixed/> (дата звернення: 20.05.2021).

					ІАЛЦ.467200.003 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

15. RealTimeLogic/LspAppMgr-ESP32. *GitHub*. URL: <https://github.com/RealTimeLogic/LspAppMgr-ESP32> (дата звернення: 27.04.2021).
16. Rye D. AV and automation industry emagazine - dave rye @ X10 | hometoys. *Wayback Machine*. URL: <https://web.archive.org/web/20140930080338/http://hometoys.com/emagazine.php?url=/htinews/oct99/articles/rye/rye.htm> (дата звернення: 17.03.2021).
17. Tsang K. F. The IOT mediated built environment: a brief survey. *Keio University*. URL: <https://keio.pure.elsevier.com/en/publications/the-iot-mediated-built-environment-a-brief-survey> (дата звернення: 12.04.2021).
18. U-blox. Nina-w10 series stand-alone multiradio modules data sheet. *arduino.cc*. URL: [https://content.arduino.cc/assets/Arduino_NINA-W10_DataSheet_\(UBX-17065507\).pdf](https://content.arduino.cc/assets/Arduino_NINA-W10_DataSheet_(UBX-17065507).pdf) (дата звернення: 26.04.2021).
19. Upton L. The Raspberry Pi in scientific research - Raspberry Pi. *Raspberry Pi*. URL: <https://www.raspberrypi.org/blog/the-raspberry-pi-in-scientific-research/> (date of access: 27.04.2021).
20. Views of caregivers on the ethics of assistive technology used for home surveillance of people living with dementia / M. Mulvenna та ін. *Neuroethics*. 2017. Т. 10, № 2. С. 255–266. URL: <https://doi.org/10.1007/s12152-017-9305-z> (дата звернення: 14.04.2021).
21. Wallace M. Fragmentation is the enemy of the internet of things. *Qualcomm.com*. URL: <https://www.qualcomm.com/news/onq/2016/02/19/fragmentation-enemy-internet-things> (дата звернення: 13.04.2021).

					ІАЛЦ.467200.003 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

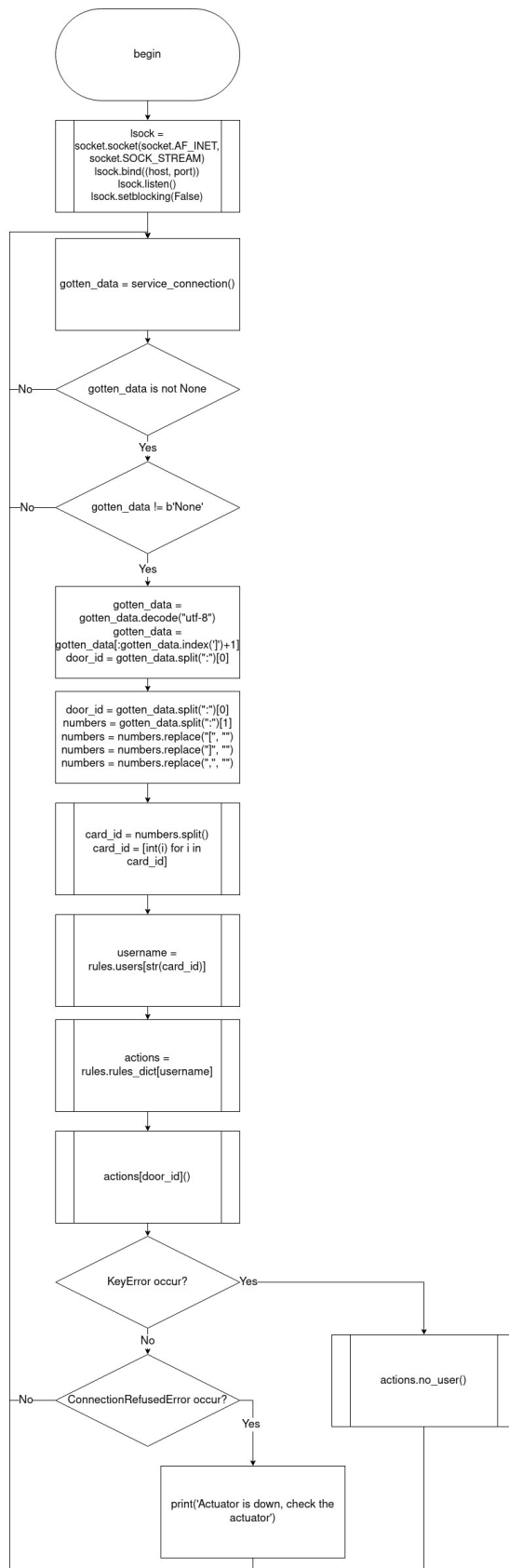
22. WiFi модуль Arduino: варианты устройств, подключение. *АрдуиноПлюс*.

URL: <https://arduinoplus.ru/podkluchenie-wi-fi-modulya-k-arduino/> (дата
звернення: 26.04.2021).

					ІАЛЦ.467200.003 ПЗ	Арк.
						77
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Додаток А
до дипломного проекту
на тему: «Система тихого розумного будинку»

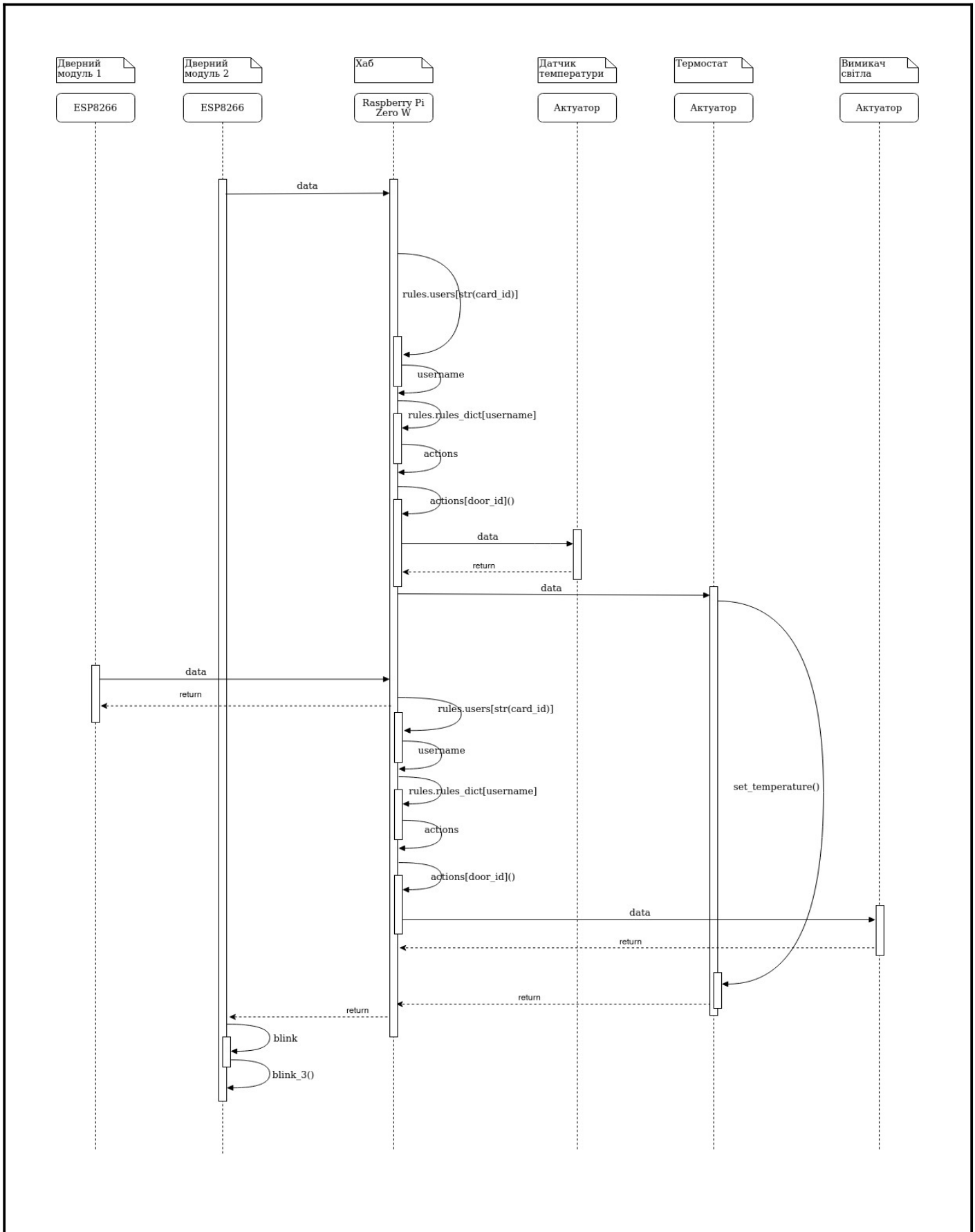
Київ – 2021 рік



					ІАЛЦ.467200.004 Д1						
Змн.	Арк.	№ документа	Підпис	Дата	Система тихого розумного будинку Принципова схема			Літ.	Аркуш	Аркушів	
Розроб.	Єрмоленко В. Р.								1	1	
Перев.	Аленін О. І.							КПІ імені Ігоря Сікорського, ФІОТ, Група ІО - 71			
Реценз.											
Н. Контр.	Сімоненко В.П.										
Затверд.	Стіренко С.Г.										

Додаток Б
до дипломного проекту
на тему: «Система тихого розумного будинку»

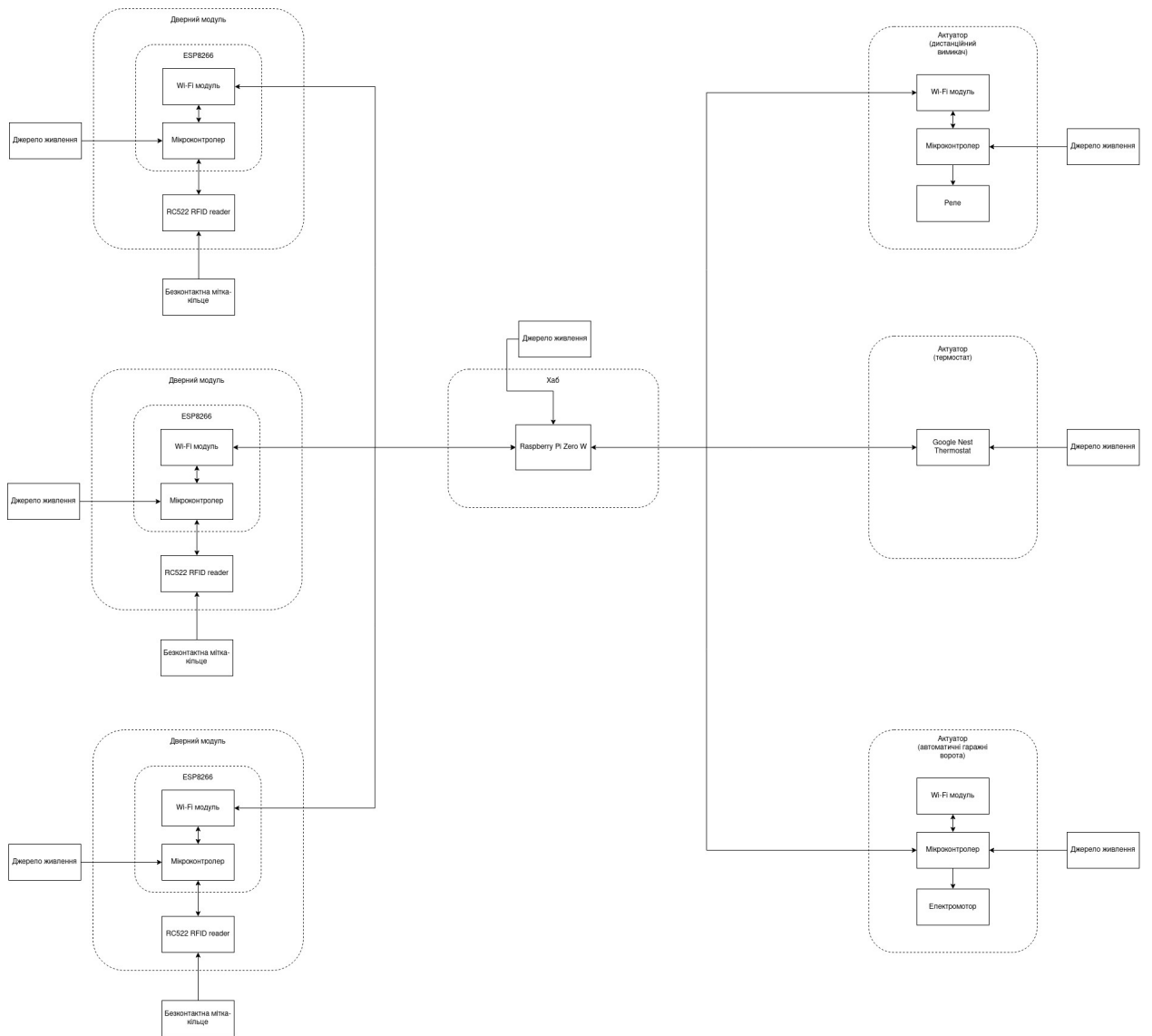
Київ – 2021 рік



					ІАЛЦ.467200.005 Д2							
Змн.	Арк.	№ документа	Підпис	Дата	Система тихого розумного будинку Функціональна схема			Літ.	Аркуш	Аркушів		
Розроб.		Єрмоленко В. Р.								1	1	
Перев.		Аленін О. І.						КПІ імені Ігоря Сікорського, ФІОТ, Група ІО - 71				
Реценз.												
Н. Контр.		Сімоненко В.П.										
Затверд.		Стіренко С.Г.										

Додаток В
до дипломного проекту
на тему: «Система тихого розумного будинку»

Київ – 2021 рік



					ІАЛЦ.467200.006 ДЗ		
Змн.	Арк.	№ документа	Підпис	Дата			
Розроб.	Єрмоленко В. Р.				Літ.	Аркуш	Аркушів
Перев.	Аленін О. І.					1	1
Реценз.					КПІ імені Ігоря Сікорського, ФІОТ, Група ІО - 71		
Н. Контр.	Сімоненко В.П.						
Затверд.	Стіренко С.Г.						
Система тихого розумного будинку							
Структурна схема							