

Міністерство освіти і науки України  
Департамент освіти і науки виконавчого органу Київської міської ради  
(Київської міської державної адміністрації)  
Київське територіальне відділення малої академії наук України  
Комунальний позашкільний навчальний заклад  
«Київська мала академія наук учнівської молоді»

Відділення Комп'ютерних наук  
Секція Технології програмування

РОЗРОБКА ТА ВИКОРИСТАННЯ АЛЬТЕРНАТИВНОГО ПРОТОКОЛУ  
ЗВ'ЯЗКУ В СИСТЕМАХ РОЗУМНОГО ДОМУ З ЕЛЕМЕНТАМИ ГОЛОСОВОГО  
АСИСТЕНТА

РОБОТУ ВИКОНАВ:  
Яценко Євгеній Ігорович  
16.03.2005 р.н.  
учень 9 класу  
Ліцею №303 Дарницького району  
Педагогічний керівник:  
Свистун Юрій Анатолійович  
вчитель інформатики ліцею №303  
Науковий керівник:  
Шмалій Григорій Григорович, студент  
Фізико-технічного інституту  
Національного технічного  
університету України «Київський  
політехнічний інститут імені Ігоря  
Сікорського»

# Комунальний позашкільний навчальний заклад «Київська Мала академія наук учнівської молоді»

## Анотація

### РОЗРОБКА ТА ВИКОРИСТАННЯ АЛЬТЕРНАТИВНОГО ПРОТОКОЛУ ЗВ'ЯЗКУ В СИСТЕМАХ РОЗУМНОГО ДОМУ З ЕЛЕМЕНТАМИ ГОЛОСОВОГО АСИСТЕНТА



Яценко Євгеній Ігорович,  
учень 9 класу,  
Ліцей №303 м. Києва

Науковий керівник: Шмалій Григорій Григорович, Фізико-технічного інституту Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»

Дослідницьку роботу присвячено розробці альтернативного протоколу бездротового зв'язку та виготовленню на його основі діючого зразка системи «Розумний дім» з елементами голосового асистента на основі загальнодоступних компонентів.

Досліджено основні аспекти та питання понять «радіозв'язок», «голосовий асистент», «розумний дім».

Проведено аналіз популярних систем «розумного дому» таких як: Ajax Systems, Broadlink, Xiaomi, Sonoff. Проаналізовані популярні протоколи бездротового зв'язку такі як: Z-Wave, ZigBee, Jeweller та переваги, можливості та ефективність використання систем «розумного дому».

Практична частина даного дослідження присвячена створенню альтернативного протоколу бездротового зв'язку під назвою M.S.P та виготовленню, на його основі дослідного зразка системи «розумний дім» з використанням мікроконтролерів ATMEL ATmega328P та ESPRESSIF ESP8266-12E. Під час проведення тестувань альтернативного протоколу зв'язку та системи «розумний дім» було визначено, що зв'язок зберігається навіть через 3 бетонні стіни, а максимальна швидкість передачі даних сягає 2 Мбіт за секунду. В результаті було розроблено альтернативний протокол зв'язку для системи «розумний дім», котрий може конкурувати з аналогами на ринку та надає змогу покращити якість передачі даних без великих затрат ресурсів.

**Ключові слова:** система «розумний дім», радіозв'язок, голосовий асистент, процедурне програмування, мікроконтролери.

## ЗМІСТ

<b>ВСТУП</b>	<b>4</b>
<b>РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ МОНІТОРИНГУ ТА УПРАВЛІННЯ ДОМОМ</b>	<b>6</b>
1.1 Технології систем «розумний дім»	6
1.2. Історія створення системи «розумного дому»	8
1.3. Історія радіо зв'язку	9
1.4. Історія розвитку голосових асистентів	10
1.5. Функціональність сучасних систем «розумного дому»	11
1.6. Аналіз популярних в Україні систем «розумний дім»	12
1.7. Аналіз протоколів радіозв'язку	14
1.8. Доцільність систем «розумний дім»	16
<b>ВИСНОВКИ ДО РОЗДІЛУ 1</b>	<b>18</b>
<b>РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ «Розумний дім»</b>	<b>19</b>
2.1. Вибір апаратної частини	19
2.2. Протокол зв'язку	21
2.3. Інтерфейс системи	22
<b>ВИСНОВКИ ДО РОЗДІЛУ 2</b>	<b>23</b>
<b>РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ ТА ОТРИМАНІ РЕЗУЛЬТАТИ</b>	<b>24</b>
3.1. Апаратна частина	24
3.2. Веб-сервер, файловий сервер та веб-інтерфейс	25
3.3. Програмне забезпечення сенсорів	26
3.4. Протокол зв'язку у головному модулі	27
3.5. Тестування та результати	29
<b>ВИСНОВКИ</b>	<b>31</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>33</b>
<b>ДОДАТКИ</b>	<b>35</b>

## ВСТУП

**Актуальність і доцільність теми.** Поняття “розумний дім” означає, що будівля обладнана таким чином, щоб всі елементи (побутові прилади, різноманітні датчики тощо) могли б «контактувати» один з одним з максимальною ефективністю забезпечення функціоналу, а їх обслуговування було б організовано раціональним чином. «Розумний дім» має масу переваг і дозволяє власникам дистанційно керувати побутовими приборами, системами опалення, водопостачання, системами безпеки, тощо, а голосовий асистент додасть ще більшої зручності та комфорту в користуванні системою, особливо для людей з особливими потребами.

Проте більшість систем «розумного дому» мають високу вартість, що є мабуть їх головним недоліком. Є і відносно недорогі системи розумного дому проте вони мають малий радіус дії, часом не зрозумілий інтерфейс, часом можуть не коректно працювати.

**Мета.** Розробити протокол бездротового зв'язку та виготовити дослідний зразок системи «розумний дім» з елементами голосового асистента на основі загальнодоступних компонентів.

**Об'єкт досліджень.** Процес розробки та створення протоколів бездротового зв'язку.

**Предмет досліджень.** Розробка протоколу бездротового зв'язку та виготовлення дослідного зразка системи «розумний дім» з елементами голосового асистента на основі загальнодоступних компонентів.

**Прикладна значущість отриманих результатів.** Досліджено відомі протоколи бездротового зв'язку (Z-Wave, ZigBee, Jeweller), системи розумного дому (Ajax Systems, Broadlink, Sonoff, Xiaomi). Створено альтернативний протокол зв'язку котрий може покращити сучасні системи «розумного дому», а саме підвищити швидкість та пропускну здатність передавання даних без великих витрат ресурсів. На основі альтернативного протоколу зв'язку виготовлено дослідний зразок. Для системи був розроблений веб-інтерфейс, що дозволяє керувати нею майже з будь-якого пристрою, на якому є веб-браузер та доступ до мережі Інтернет. Також були проведені тестування створеного протоколу зв'язку на загальну проникаючу здатність,

швидкість передачі даних та роботи дослідної системи «розумний дім». Результатом стало те, що максимальна швидкість передачі даних становить 2 МБіт за секунду, відправка або прийняття сигналу займає близько 700 мікросекунд, а проникаюча здатність склала 3 бетонні стіни.

**Методи дослідження.** Використано такі методи дослідження: аналіз, синтез, індукція, експеримент.

## РОЗДІЛ 1

### КЛАСИФІКАЦІЯ (ОГЛЯД) ІСНУЮЧИХ СИСТЕМ МОНІТОРИНГУ ТА УПРАВЛІННЯ ВУЗЛАМИ РОЗУМНОГО ДОМУ.

#### 1.1. Технології систем «розумний дім»

«Розумний дім» — система домашніх пристроїв, здатних виконувати дії для забезпечення життєдіяльності будинку без участі людини. Система «розумного дому» — це взаємопов'язані між собою пристрої, датчики, прилади тощо. Тобто – це певна кількість модулів, об'єднаних в одну мережу, де кожний модуль виконує певну функцію, що в свою чергу може допомагати в контролюванні параметрів житлового будинку (вмикати/вимикати чайник, світло, обігрівач), у відстежуванні енергоспоживання оселі, в підтримці температури повітря в будівлі тощо. Головна перевага заключається в тому, що системи «розумний дім» дозволяють не тільки бачити поточні параметри житла такі, як температура повітря, відносна вологість та інших, а й надають можливість контролювати ці параметри в реальному часі, наприклад увімкнення/вимкнення світла або електроприладів. Всі ці параметри можна переглядати у різний спосіб, деякі системи розумного дому використовують невеликі дисплеї — консоль або фізичні кнопки на модулях системи, інші використовують веб-інтерфейс або мобільний додаток. Найбільш популярними способами керування є веб-інтерфейс та мобільний додаток, оскільки за допомогою них можна керувати системою з будь-якої точки світу де є мережа Інтернет і мобільний зв'язок. Також в мобільних додатках для систем «розумного дому» набирає популярності голосовий асистент, оскільки він може пришвидшити виконання деяких функцій системи, наприклад замість того, щоб власноруч шукати потрібний сценарій або функцію, потрібно лише сказати назву цієї функції і система сама «все знайде і виконає». На сьогоднішній день в світі існують багато систем «розумного дому». Вони поділяються на:

- **Централізовані** — мають один головний модуль, котрий збирає, опрацьовує всі дані від сенсорів системи і відображає вже опрацьовані дані. Переваги таких

систем: відносно недорогі в виготовленні, їх легше програмувати. Проблеми заключаються у тому, що, під час виходу з ладу головного модуля системи, вся система перестане працювати.

- **Децентралізовані** — кожний модуль може бути головним, тобто збирати та опрацьовувати дані незалежно. Переваги таких систем полягають в тому, що розв'язується головна проблема централізованих систем, а саме стійкість до виходу з ладу головного модуля системи. Проблеми ж заключаються в тому, що такі системи більш складно програмувати, налагоджувати, а також їх виробництво більш вартісне (500\$ і більше), тому що використовуються потужні мікроконтролери високої вартості.

В свою чергу централізовані та децентралізовані поділяються на:

- **Дротові** — всі елементи системи з'єднуються за допомогою дроту. Перевагою такого зв'язку є швидкість обміну даними та безпека, оскільки для отримання доступу до даних потрібне фізичне приєднання до дроту. Швидкість, в свою чергу досягається завдяки тому, що передавання даних здійснюється через дріт, який максимально та перешкодо захищений. Проблеми також наявні: досить великий об'єм дроту, котрий потрібно прокласти у помешканні, а це потребує і додаткового ремонту.
- **Бездротові** — всі елементи системи з'єднуються за допомогою радіо зв'язку. Перевагою таких систем є те, що їх можна встановлювати навіть після ремонту, оскільки фізичного з'єднання немає, що дозволяє використовувати елементи системи в межах помешкань майже будь-де. Проблемою такого з'єднання є те, що радіозв'язок має свої межі дії, і найчастіше ці межі можуть задовольнити лише господарів квартир та власників невеликих приватних будинків.

Сама система «розумний дім» найчастіше складається з таких елементів, які схематично зображені у Додатку А:

1. Шлюз — це програмно-апаратний засіб, який дозволяє системі обмінюватися даними з серверами компанії або напряму з системою показу цих даних (інтерфейс, консоль, мобільний додаток).

2. Головний модуль — збирає всі дані з сенсорів, опрацьовує їх і в результаті може робити деякі дії (вмикати світло, регулювати систему опалення тощо).
3. Сенсори — це допоміжні модулі, які контролюють стан та функціональність об'єктів. На сенсорах розташовані різні датчики відповідно до профілю, наприклад сенсор температури-вологи обладнаний датчиком для виміру температури та відносної вологості повітря. Таких сенсорів є безліч, для прикладу сенсори виявлення чадного газу, сенсори виявлення протікання води і багато інших сенсорів [2, 24, 25].

## 1.2. Історія створення системи «розумного дому»

Перші спроби керування чимось на відстані можна віднести до відомого у всі часи винахідника Ніколи Тесли, який у 1898 р. розробив систему дистанційного керування кораблями й іншими транспортними засобами. Вже у 1915-1920 роки почали з'являтися перші електричні прилади. Популярність електричних приладів стрімко зростала і все більше людей почали їх використовувати. Оскільки прогрес невинно крокував, люди вже потребували автоматизації багатьох процесів своєї діяльності, додати їм певну «інтелектуальність» для більш ефективної роботи. Так, вперше, щось схоже на систему «розумного дому» було представлено у 1934 році на ярмарках у Чикаго та Нью-Йорку.

Розвиток електроніки продовжувався, і в 1960-х роках ряд інновацій в апаратному та програмному забезпеченні дозволили людям мати доступ до перших комп'ютерів в своїх домівках. На основі чого, перша система, що нагадувала систему «розумний дім» з'явилася вже у 1966 р. Це була експериментальна система автоматизації домашнього побуту — «Electronic Computing Home Operator IV», проте це був приватний та некомерційний проект. Його винахідником був інженер компанії Westinghouse Electric — Джим Сазерленд. Спочатку система була створена для автоматизація сімейних фінансів, потім функціонал був розширений, тож згодом вона могла контролювати температуру помешкання та побутові прилади, але головним його недоліком були величезні розміри, котрі вона займала у приміщенні.



Далі саме створення та розповсюдження мікроконтролерів, транзисторів й інших напівпровідникових елементів витіснило об'ємні схеми на радіолампах й зробило процес розробки будь-яких електронних приладів значно простішим, швидшим й дешевшим. Деякі сьгоднішні системи розумного дому дозволяють збирати дім компонентно та за певними вподобаннями і потребами власників поіешкань, тобто процес добору модулів та сенсорів має адаптивний підхід.

Тепер майже будь-який побутовий прилад можна приєднати до систем «розумного дому» для виконання потрібних функцій (контроль опалення, дистанційне керування духовою шафою, вмикання електроприладів тощо) [2].

### 1.3. Історія радіозв'язку

**Радіо** — загальна назва методів бездротового передавання та отримання інформації за допомогою електромагнітних хвиль радіочастотного діапазону.

Обмін інформацією засобами радіо здійснюють у багатьох сферах нашого життя, наприклад: у системах радіозв'язку (рації, побутове радіо), у телебаченні, у радіокеруванні різними пристроями та системами, в активній та пасивній радіоіндетифікації (RFID) тощо.

Питання першості відкриття радіо досі однозначно не встановлено, бо розвиток радіо був досить довгий. У 1878 році англійським вченим Джеймсом Кларком Максвеллом була сформульована теоретична частина радіо. Практичну частину виконав у 1886 року німецький вчений Генріх Герц, який побудував передавач і приймач радіохвиль та дослідив їх фізичні властивості. Це стало поштовхом для подальшого дослідження та використання радіо. Подальші дослідження вели: Нікола Тесла в Америці, Олівер Лодж, Гульєльмо Марконі в Європі, Микола Пильчиков, Семен Айзенштейн, Сергій Жидковський в Україні та інші. Вже у 1904 році Джон Амброс Флемінг винайшов електровакуумну лампу, котра виконувала функцію детектора. Але практично їх почали використовувати лише коли фізики зрозуміли, що тріод (електровакуумна лампа з 3 контактами) можна використовувати для підсилення звуку. Перша передача звуку відбулася завдяки

Реджиналду Фессендену вже у 1906 році, а в 1920 році електровакуумні лампи почали використовувати у радіо приладах. У тогочасних винахідників було багато спроб створити прилад для передачі звуку у два напрями, щось на зразок телефона, проте були й винахідники котрі хотіли транслювати передачі на широку публіку.

Так широке радіомовлення розпочав Чарлз Герролд у 1909 році в Каліфорнії [8].

#### 1.4. Історія розвитку голосових асистентів

**Голосовий асистент** — програма або сервіс, що дає можливість розпізнавати голосові команди від людини та відповідати на них. Розпізнавання та синтез мови виконується за допомогою різних математичних алгоритмів.

Першим електронним пристроєм для синтезу мовлення став Voder, котрий був розроблений Гомером Дадлі у 1937 в стінах компанії Bell. Перший розпізнавач мови був розроблений в 1952 компанією Bell, пристрій мав назву Audrey і міг розпізнавати числа від 1 до 9. 1990 року була створена перша масова програма Dragon Dictate, після чого в 1994 році IBM представила свою розробку — ViaVoice. 1995 року була представлена телефонна система Val від компанії Bell, котра виконувала автоматизацію роботи диспетчерів для маршрутизації телефонних дзвінків. Історія голосових асистентів не припиняється і досі, кожного дня голосові асистенти стають все більш функціональнішими.

Серед сучасних голосових асистентів найтехнологічніші:

- Siri — від компанії Apple. Siri розробляється з грудня 2007 року. Спочатку Siri була доступна в App Store, як додаток для iOS від Siri Inc. Після придбання компанії Siri Inc. компанією Apple, у квітні 2010 року, Siri стала невід'ємною частиною операційної системи iOS, починаючи з п'ятої версії.
- Google Assistant — від компанії Google LLC. Вперше був показаний у 2016 році на конференції Google I/O 18. В серпні 2018 року Google Assistant частково запрацював в Україні, проте його введення призупинили через не повну

підтримку української мови, проте згодом додаток зміг повністю підтримувати українську мову.

Такі сучасні голосові асистенти можуть виконувати значну кількість задач, наприклад: дати рекомендації як одягатися, враховуючи погодні умови, сповістити про курс валют, шукати інформацію в мережі інтернет, розповідати казки, викликати таксі та багато чого іншого [6, 7, 10].

## **1.5. Функціональність сучасних систем «розумного дому»**

Технології постійно розвиваються і покращуються. Всі сфери автоматизації покращуються (машинобудування, маніпулятори тощо) і системи розумного дому не виключення. Системи розумного дому дозволяють створювати сценарії, в яких користувач зазначає, що має робити система при тих чи інших умовах та ситуаціях. Отже зазначимо сценаріїв що можуть допомагати людині:

- контроль доступу до приміщення;
- облік та контроль багатьох параметрів, ресурсів і швидке реагування на різні, в тому числі і непередбачувані людиною ситуації, до того ж реакція є комплексною і миттєвою;
- віддалений контроль і керування будівлею, тому що всі інформаційні та керуючі канали зв'язку в такій системі є цифровими та інші.

Лише дотиком до консолі або до екрану телефона можна увімкнути сценарій, який виконає дії, що були запрограмовані. Для прикладу, сценарій «Я вдома» міг би увімкнути світло, встановити комфортний мікроклімат, закрити штори, увімкнути улюблений канал на телевізорі та інші. Можна керувати різними побутовими приладами, наприклад за допомогою кухонної плити або мультиварки розігріти вечерю або заздалегідь встановити час приготування вечері мультиваркою, випрати білизну тощо. Автоматичне ввімкнення чи вимкнення зовнішнього освітлення залежно від часу доби або освітленості не тільки забезпечить комфорт, а й може зробити ефект присутності господаря в помешканні і тим самим відлякати злодіїв. Системи можуть постійно вимірювати температуру й підтримувати її в кожній

кімнаті на заданому рівні, керуючи електроклапанами труб з теплою водою або заслінками кондиціонера, а також при піднятті рівня CO<sub>2</sub> зможе автоматично ввімкнути/вимкнути вентиляцію. А якщо взяти до уваги те, що за допомогою індивідуальних сценаріїв можна економно використовувати споживання електроенергії та інших енергоресурсів кожен день, то не стає зрозуміло, що системи «розумний дім» допомагають спростити не тільки побут, а й ефективно використовувати ресурси. Також не менш важлива і безпека помешкання. «Розумний дім», у разі відсутності власників, може «зустріти» злодіїв гучною сиреною і швидким викликом патрульної поліції. Або, у разі витoku газу або води, може повідомити не тільки господаря, а й відповідні аварійні служби та створити звіти у мобільному додатку про всі події, які відбувалися.

## 1.6. Аналіз популярних в Україні систем «розумний дім»

- **Ajax Systems** — міжнародна компанія. Розпочала свій шлях у 2011 році. Головний штаб та лінії виробництва розташовані у Києві. Оскільки компанія розпочала свій шлях в Україні, то система підтримується нашими охоронними фірмами. Ця система розумного дому виконує, як і функцію забезпечення комфорту та зручності в управлінні житлом, так і гарантує його безпеку за допомогою багатьох сенсорів, наприклад: сенсор руху, сенсор розбиття скла та ін. Обладнання Ajax працює на зашифрованому та захищеному каналі радіозв'язку на протоколі «Jeweller», який є власною розробкою Ajax Systems. Усі сенсори бездротові, тобто мають власний елемент живлення, головний хаб працює від мережі 220В, а завдяки резервному акумулятору хаб може працювати деякий час без напруги 220В [18].

Переваги:

- Швидкий та простий монтаж.
- Велика зона покриття сигналу(до 2000 м.).
- На високому рівні реалізована безпека як і сенсорів так і системи, до речі на сенсорах є навіть захист від зняття його зі стіни.

- Значна кількість каналів зв'язку хаба з фірмами охорони та власниками (LAN, Wi-Fi, GSM).
- Різні способи інформування користувача (дзвінок, Push-повідомлення, SMS повідомлення).

Недоліки:

- Відносно висока вартість системи (більше 216\$ за базовий комплект).
  - Система працює лише тоді, коли хаб працює.
- **Broadlink** – Китайська компанія. Через це, за замовчуванням в продукції цієї компанії немає підтримки української мови. Обладнання Broadlink представляє комплект сучасних гаджетів. Кожен елемент системи може працювати окремо або ж у єдиній системі. Наявні такі модулі: хаб, сенсор чистоти повітря, смарт розетки, сенсори руху, освітлення, шуму та інші сенсори та системи [19].

Переваги:

- Досить швидкий монтаж.
- Великий вибір сенсорів.
- Сенсори працюють, як з хабом так і без нього.

Недоліки:

- Незначна відстань покриття сигналу (50 м на відкритій місцевості).
  - Відсутня резервна батарея у головного хаба.
- **Sonoff** — це лінійка продуктів для систем «Розумний дім» від компанії Itead Studio. Спеціалізація цієї компанії полягає у виробництві високотехнологічної сучасної електроніки і простих пристроїв Internet of Things. Устаткування Sonoff направлено на побудову систем домашньої автоматизації з бюджетних, багатофункціональних і «розумних» пристроїв з простою і доступною кожному встановленням і налаштуванням. Але датчики цієї системи не дуже популярні, найчастіше власники використовують лише виконавчі пристрої на базі реле, що унеможливорює розробити централізовану систем [20].

Переваги:

- Досить дешевий (від 250 гривень.).
- Не має хабу, кожний пристрій досить самостійний пристрій.

- Багатий асортимент виконавчих пристроїв, але сенсори не здобули масштабної популярності.

Недоліки:

- Мала популярність сенсорів для системи sonoff.
- Залежить від роботи серверів компанії.
- **Xiaomi** — Китайська компанія. Виходячи з того, що компанія китайська, отримуємо деякі проблеми з українською мовою, а саме її повної відсутності. Система розумного дому від Xiaomi є бюджетною, тому є простою у використанні з чого випливає, що кожна людина може легко зробити свій дім або квартиру «розумною». Елементи системи працюють як з хабом, так і без нього. Велика кількість як і сенсорів, так і виконуючих пристроїв, крім того ця система підтримує прилади інших компаній.

Переваги:

- Повна автономність пристроїв.
- Є камери власного виробництва.
- Невисока вартість.

Недоліки:

- Недостатня відстань покриття сигналу (до 10м).
- Відсутність резервного живлення у хаба.
- Проблеми з перекладом програмного забезпечення українською мовою.

## 1.7. Аналіз протоколів радіозв'язку

**Протокол зв'язку** — це набір стандартних правил представлення даних, ідентифікації, виявлення помилок та передачі сигналів, необхідних для обміну даними.

На сьогоднішній день протоколів радіо зв'язку досить багато. Серед цих протоколів визначити можна такі:

- **Z-Wave** — це протокол бездротового зв'язку, котрий використовується переважно для домашньої автоматизації, тобто саме для таких пристроїв, котрі

виконують короткі команди. Зазвичай використовується в частотах до 1.1ГГц ( у Північній Америці використовують частоту 908.42МГц) [9].

- Переваги:
  - Розроблений саме для домашньої автоматизації.
  - Низьке енергоспоживання.
  - Має здатність до самовідновлення у разі виходу з ладу якогось вузла.
- Недоліки:
  - Невисока швидкість передачі даних в 100 кбіт на секунду.
  - Невисока проникаюча здатність в реаліях міської забудови.
- **ZigBee** — бездротовий протокол передачі даних, котрий заснований у 2002 році. Протокол заснований однойменним альянсом ZigBee™. Працює ZigBee на протоколі IEEE 802.15.4. Зараз існують різні версії протоколу, що називаються однойменно з роком їх створення. ZigBee 2004 — перша специфікація з базовим функціоналом, випущена в 2004 році. Через два роки, була розроблена друга версія — ZigBee 2006. А ще через рік у 2007 році була розроблена версія ZigBee 2007 та ZigBee Pro. Наразі найбільш часто використовувана версія протоколу ZigBee — це ZigBee 2007 [8].
  - Переваги:
    - Низьке енергоспоживання.
    - Відносно великий радіус дії, тобто має гарну проникну здатність.
    - Швидкий вихід зі сплячого режиму (близько трьох мілісекунд).
  - Недоліки:
    - Невисока швидкість передачі даних 20-250 кбіт на секунду.
    - Відносно висока ціна адаптерів та шлюзів для під'єднання пристроїв ZigBee до мережі Wi-Fi.
- **Jeweller** — протокол бездротової передачі даних. Власна розробка компанії Ajax Systems. Протокол має двосторонній радіозв'язок, що працює на частотах 868,0–868,6 МГц з дальністю зв'язку до 2000 м. Вона має автоматичне регулювання потужності сигналу (до 25 мВт). Технологія передбачає

адресність пристроїв та блочне шифрування інформації з динамічним ключем. Доставлення аварійного сигналу займає не більше 0,15 секунди [18].

- Переваги:
  - Велику відстань роботи.
  - Швидке опитування сенсорів.
  - Наявні захисти від глушіння та підміни.
- Недоліки:
  - Закритий протокол(тобто доступ до розробки на цьому протоколі зв'язку є лише у компанії Ajax Systems, що покращує безпеку протоколу, проте змушує купувати обладнання лише від Ajax).
  - Має велику ціну( оскільки це власна розробка компанії, і їй самій треба постійно вкладати гроші на розвиток протоколу ).

Серед усіх протоколів за характеристиками найкращим є Jeweller. За допомогою нього можна ловити сигнал на відстані 2 км на відкритій місцевості, він має змінну потужність та систему підтримки потрібної потужності тим самим підтримуючи роботу сенсорів системи до 7 років від батарейок, та має алгоритми, котрі забезпечують захист від злому та підміни сигналів. Але найбільшою проблемою даного протоколу є те, що він доступний тільки для продукції від компанії Ajax, а це означає, що він не може бути використаний у бюджетних системах «розумний дім».

## **1.8. Доцільність системи «розумний дім»**

У багатьох країнах з досить розвинутою економікою вся система доходів та витрат платників податків є прозорою, що дозволяє правильно і ефективно працювати як економіці, так і правоохоронній системі. Це призводить до того, що придбання житла в кредит може дозволити собі більшість працездатного населення. Кожен, хто взяв кредит на купівлю житла, стане «жити в борг». Більшість доходу таких громадян піде на погашення кредитних платежів. Але власники житла стикаються і з великою кількістю ризиків та . Тому в країнах з розвинутою економікою без страхування



ризиків кредитування не відбувається. Природно, що страхові компанії зацікавлені в якості і надійності застрахованого житла. Але оскільки виникнення страхових випадків не уникнути, то страхові компанії прагнуть потім стягнути гроші зі справжніх винуватців, наприклад забудовника. «Розумний дім» може надати експертам так звану «чорну скриньку», згідно вмісту якого будуть робитися висновки. Навіть в Україні сучасні офісні і житлові будівлі вже проектуються з урахування подальшого впровадження систем «розумного дому». Також і власники цих сучасних будівель зацікавлені в «розумності» будівлі, тому що без сумніву можна сказати, що встановлення систем «розумного дому» приносить значну економію коштів за рахунок більш точного контролю над усіма системами будівлі, а також раціонального використання ресурсів.

Та не все так просто, оскільки система «розумний дім»— це комп'ютеризована система, яка може в будь-який момент не спрацювати, загубити пристрій чи просто зламатися. Що буде робити система при спробі зловмисників зламати протокол зв'язку чи перехопити або підмінити пакети у канал зв'язку, чи просто вимкнути світло у квартирі? Зрозуміло, що повністю довіряти системам «розумного дому» не потрібно, проте зараз всі виробники намагаються забезпечити хоча б мінімальний рівень безпеки, бо від стабільності роботи системи може залежати не одне життя. Проте головними проблемами цих систем залишаються складний та не зрозумілий інтерфейс, ціна, у бездротових системах слабкі за потужністю радіомодулі, наявність довгих дротів у домівці або не дуже гарна перспектива ремонту у дротових систем.

З цього й випливає потреба створити альтернативний протокол зв'язку, який буде мати кращі характеристики ніж аналоги та буде доступний для всіх систем, що зменшить вартість встановлення систем «розумного дому», матиме кращу якість передачі даних та буде захищений від кіберзлочинців. Що і є метою та актуальністю даної роботи. А на основі такого протоколу створити дієвий дослідний зразок системи «розумний дім», який дозволяє тестувати та швидко змінювати протокол, також потрібно створити веб-інтерфейс, який можна буде швидко змінювати та покращувати на відміну від сумнівних готових рішень[1].

## ВИСНОВКИ ДО РОЗДІЛУ 1

У розділі 1 розглянуто історію розумного дому, історію розвитку радіозв'язку людством, та історію голосових асистентів від перших видів до сьогодення. Проведено аналіз існуючих популярних систем «розумного дому» та популярних протоколів бездротового зв'язку. Приведені можливості сучасних систем «розумного дому» та доцільність встановлення розумного дому. Визначені недоліки існуючих як і систем «розумного дому», так і протоколів зв'язку. Знайдені можливі небезпеки при користуванні різними системами «розумного дому», а також доведена мета та актуальність роботи.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ СИСТЕМИ «РОЗУМНИЙ ДІМ»

#### 2.1. Вибір апаратної частини

Будь який проект, а тим паче система «розумний дім» має власний алгоритм розробки. Загальну схему алгоритму розробки можна побачити у Додатку Б. Спочатку потрібно проаналізувати проблематику та сучасний ринок систем «розумний дім». Після аналізу потрібно підібрати всі необхідні компоненти для побудови майбутнього пристрою. Наступним кроком слідує проектування майбутнього протоколу зв'язку, системи «розумний дім» і веб-інтерфейсу для зручного керування системою. Після практичної реалізації, а саме веб-інтерфейсу та серверної частини, програмного коду системи «розумний дім» й схем та друкованих плат потрібно провести тестування та зробити висновки. Так, будь-яка система розумного дому базується на електронних компонентах. Для дорогих систем «розумного дому» обираються плати на яких встановлені потужні контролери з різними архітектурами, наприклад: X64, X86. Дешеві ж системи, які були проаналізовані та описані в роботі, зазвичай будуються на мікроконтролерах з архітектурами: AVR, PIC, ARM. На даний момент вибір мікроконтролерів досить великий, тому обрати потрібний мікроконтролер дуже важко. Доступні та популярні серії мікроконтролерів в радіоелектроніці:

- мікроконтролери різних серій від компанії Atmel.
  - Найбільш популярна серія це Atmel ATmega.
- мікроконтролери різних серій від компанії PIC.
  - Одна з найпопулярніших серій це серія PIC12/PIC16.
- мікроконтролери різних серій від компанії STMicroelectronics.
  - У компанії STMicroelectronics також як і у компанії PIC багато популярних контролерів. Але з них можна вирізнити контролери серії STMicroelectronics STM32(з 32 бітними контролерами) та серії STMicroelectronics STM8(з 8 бітними контролерами).

- мікроконтролери різних серії від компанії Espressif Systems.
  - Серія мікроконтролерів Espressif Systems Esp8266.
  - Серія мікроконтролерів Espressif Systems Esp32(менш популярні за Esp8266).

В поданому проекті було використані різні контролери, а саме Esp8266-12E від компанії Espressif Systems та Atmega328P від компанії Atmel. Було обрано ці мікроконтролери за достатньо низьку ціну та гарні характеристики, а саме:

#### **Atmega328P:**

- 32 кБ — пам'ять програми.
- 2 кБ — оперативна пам'ять.
- 1 кБ — енергонезалежна пам'ять.
- Має технологію енергозбереження — «Micro-power» .
- Напруга живлення 5В.

#### **Esp8266-12E:**

- 4 мБ — пам'яті програми та файлової системи.
- 96-128 кБ — оперативна пам'ять.
- Підтримка Wi-fi протоколу 802.11 b / g / n;
- Напруга живлення 3,3В.

Ці характеристики дозволяють цим контролерам швидко та ефективно виконувати цілі, котрі були перед ними поставлені. Мікроконтролери Atmega328P я використовую у сенсорах своєї системи, тобто він повинен мати досить низьке енергоспоживання, швидко опитувати сенсори, а також оперативно передавати інформацію за допомогою бездротового зв'язку. Мікроконтролер Esp8266-12E я використовую для головного пристрою в системі «хабу», цей контролер має велику обчислювальну потужність для того щоб опрацьовувати багато даних з різних сенсорів. Схематичне зображення альтернативної системи «розумний дім» можна побачити у Додатку В [2, 3, 4].

## 2.2. Протокол зв'язку

Для системи «розумного дому» було вирішено використовувати один з найпопулярніших модулів бездротового радіозв'язку NRF24L01+ від Nordic Semiconductor (Додаток Г). На відміну від модулів з частотою роботи в 433 МГц даний модуль має багато переваг:

1. Кожен модуль може бути як і передавачем, так і приймачем.
2. Має змінну потужність -18, -12, -6, 0 dBm.
3. Має змінну швидкість передачі даних: 0.25, 1, 2 Мбіт на секунду.
4. Має 126 незалежних каналів.
5. Розмір одного пакету 32 байта.
6. Є режим передачі в відповідь на прийом.
7. Є режим автоматичних повторів відправки пакету, якщо той не був отриманий.
8. Модуль апаратно перевіряє пакет на цілісність.

Проте, модуль має один недолік, він одночасно контактує лише з 5 іншими модулями. Тобто побудувати систему з багатьма сенсорами не можливо, якщо не подолати це обмеження. Тобто потрібно розробити протокол, котрий може ліквідувати головний недолік таких модулів. З усіх вище перелічених характеристик NRF24L01 раціональніший за модуль з частотою в 433 МГц.

Альтернативний протокол має назву M.S.P (Multiple Sensors Protocol). M.S.P вирішує одну з головних проблем цього модуля, а саме можливість одночасно контактувати лише з 5 модулями.

Згідно Додатку Д використовуються такі канали радіомодуля:

1. Для відправки запиту на отримання всіх даних сенсора, або для відправки запиту на зміну деяких параметрів сенсора.
2. Для прийняття аварійних повідомлень від сенсорів, та подальшого швидкого реагування на ці дані.

3. Для пошуку нових сенсорів, їх реєстрації та подальшого присвоєння персонального адресу для відправки запитів на отримання даних або для відправки запиту на зміну параметрів сенсора.
4. Для отримання відповідей від сенсорів на які були відправлені запити на отримання інформації або на зміну якогось параметра.

В результаті було отримано двосторонній бездротовий радіо протокол, з відносно гарною проникаючою здатністю (з бетонні стіни) та швидким опитуванням сенсорів. Для підсилення проникаючих здібностей системи можна використовувати модуль NRF24L01+ з підсилювачем та антеною (Додаток Е) [12, 13, 14].

### 2.3. Інтерфейс системи

Для розробленої системи створено альтернативний веб-інтерфейс для управління системою через будь-який пристрій з веб браузером (Додаток Є).

У цьому інтерфейсі є декілька вкладок з інформацією:

1. «Розумний Дім» — це головна вкладка на якій відбувається керування основними параметрами вашої домівки, наприклад: статуси сенсорів, управління реле та інше.
2. «Приміщення» — на цій вкладці будуть виведені дані сенсорів кімнати котру ви вибрали для перегляду.
3. «Сенсори» — на ній будуть доступні різні налаштування та службова інформація від сенсорів, наприклад: заряд батареї, потужність сигналу та інша.
4. «Сценарії» — на цій вкладці користувач може налагодити дії автоматичних сценаріїв, наприклад: течія води, різка зміна температури.

Також користувач може розробити власні сценарії для роботи системи.

Проте, крім веб-інтерфейсу буде доступний додаток для смартфонів з операційною системою на базі Android. Додаток буде мати більшим розширений функціонал з голосовим асистентом. Голосовий асистент за командою зможе: вимкнути або увімкнути світло, доповісти про мікроклімат у домівці, запустити

раніше створений сценарій або створити новий та багато чого іншого. Голосовий асистент знаходиться у розробці, розробляється на базі сервісів Google(Google Cloud), тому що Google дає безкоштовний сервіс для синтезу мови та її розпізнавання.

## **ВИСНОВОК ДО РОЗДІЛУ 2**

У розділі 2 була проаналізована робота з проектування системи «розумний дім» з використанням альтернативного протоколу бездротового зв'язку M.S.P., а саме було розглянуто та описано схему алгоритму розробки протоколу зв'язку та системи «розумний дім». Було вирішено питання щодо вибору апаратної частини проекту - вибір пав на недорогі мікроконтролери, котрі мають непогані характеристики та досить популярні у колі радіолюбителів. Вибрані мікроконтролери й інші компоненти досить легко доступні в Україні. Також був розглянуте програмне питання проекту, а саме протокол зв'язку, й схему його роботи, а також веб-інтерфейс системи, його можливості та можливості голосового асистенту у мобільному додатку.

## РОЗДІЛ 3

### ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ ТА ОТРИМАНІ РЕЗУЛЬТАТИ

#### 3.1. Апаратна частина

Проект розроблений на платах Arduino Nano та NodeMCU. Серце плати Arduino Nano мікроконтролер ATMEGA328P, а серце плати NodeMCU мікроконтролер ESPRESSIF ESP8266-12E. Всі сенсори та виконавчі пристрої розроблені з використанням плат Arduino Nano, а головний модуль системи виконаний на платі NodeMCU. Схеми розроблені у середовищі EasyEDA. На сьогодні розроблено один сенсор та один виконавчий прилад, а саме сенсор температури і вологості та виконавчий пристрій котрий вмикає чи вимикає своє реле, яке в свою чергу керує напругою в 220 Вольт. У Додатку Ж.1 подано схему сенсора температури та вологи. Згідно цієї схеми можна розробити досить дрібну друковану плату. Цей сенсор найкраще підходить для кімнатного використання оскільки вміє виміряти температуру повітря в діапазоні 0-50 градусів за шкалою Цельсія з похибкою в +/- 2 градуси, та відносну вологість повітря, в діапазоні 20-95% з похибкою в +/- 5% (Подано в Додатку Ж.2). Схему другого сенсора подано в Додатку К.1. Сенсор, навіть в базовій версії має 8 виходів для з'єднання з звичайним реле або ж твердотільним реле. Цей модуль можна використовувати будь-де, оскільки він має майже нескінченну кількість каналів управління. Їх нескінченність забезпечує мікросхема 74НС595N, на схемі позначається як U2. Ця мікросхема переводить послідовний сигнал у паралельний. Для цього використовуються 3 виходи мікроконтролера. У мікросхемі крім 8 виходів, ще є вихід послідовного під'єднання таких же мікросхем, тобто використовуючи 3 ніжки мікроконтролера ми можемо отримати 8, 16, 24... виходів які можна з'єднати з реле, транзисторами, та іншими складовими (Показано у Додатку К.2). Головний модуль має найпростішу схему (Показано у Додатку Л.1), тому що складається з двох компонентів плати NodeMCU та радіомодуля NRF24L01+, проте найскладніше програмне забезпечення, оскільки



він керує всією системою, а також забезпечує безпеку житла. Готовий пристрій головного модуля подано у Додатку Л.2[11,12,13,14,15]

### **3.2. Веб-сервер, файловий сервер та веб-інтерфейс**

Частиною програмного забезпечення головного модуля є файлова система. У файловій системі знаходяться всі допоміжні файли для роботи системи, а саме такі , як:

- Файли веб-сторінок для веб-інтерфейсу (.html).
- Файли стилей для веб-інтерфейсу (.css).
- Файли скриптів для правильної роботи веб-інтерфейсу(.js).
- Файл конфігурації для правильної роботи головного модуля(.json).

Оскільки під час процесу налагодження коду потрібно постійно вносити зміни до файлів, які знаходяться у файловій системі, а робити це постійною прошивкою плати не доцільно, виникла необхідність встановити FTP сервер на головний модуль системи. FTP сервер дозволяє мати досить легкий доступ до файлової системи головного модуля, а отже й робити швидкі зміни в файлах без перепрошивки всієї плати. FTP є дуже раціональним рішенням, проте тільки на час налагодження коду, оскільки безпека цього протоколу досить сумнівна, що може збільшити вразливість всієї системи, тому в кінцевих версіях системи планується відмовитися від використання файлового сервера FTP.

На платі головного модуля окрім файлової системи та файлового сервера знаходиться й веб сервер котрий використовує файли з файлової системи, а саме файли .html, .css, .js щоб відобразити користувачу веб-інтерфейс. Тобто, використовуючи вбудовану бібліотеку «ESP8266WebServer.h» на платі запускається HTTP сервер. Внутрішня процедура роботи веб-серверу має такий вигляд:

1. HTTP запит від клієнта на IP адресу головного модуля.
2. Доступ до файлової системи та передача контенту сторінок та інших файлів за допомоги HTTP протоколу.

3. Відображення контенту у браузері клієнта та виконання скриптів на стороні клієнта.

Візуальна частина веб-інтерфейсу (зображена у Додатку Є) написана з використанням мови гіпертекстової розмітки (html) та каскадних таблиць стилів (css). Написана вона у спеціальному редакторі коду, а саме Brackets. Серверна частина веб-інтерфейсу розроблена з використанням мови програмування javascript у тому ж редакторі, у скриптах містяться функції асинхронного опитування бази на предмет раніше отриманих даних від сенсорів та їх подальшого виведення у веб-інтерфейс. [16,17]

### 3.3. Програмне забезпечення сенсорів

Програмне забезпечення сенсорів написано у редакторі Arduino IDE, з використанням мови програмування C++. Програмне забезпечення для сенсорів має такі функції:

1. **Максимальне енергозбереження.** Оскільки планується, що сенсори матимуть живлення від батарейок, то вони мають економити енергію з максимальною ефективністю. Вже зараз є апаратні пристосування для подальшого енергозбереження, наприклад всі сенсори мають підключену ногу IRQ радіомодуля до нульового апаратного переривання мікроконтролера Atmel ATmega328. На нозі IRQ радіомодуль буде пускати низький логічний сигнал, коли буде приймати дані, який буде показником для мікроконтролера що потрібно вийти з режиму сну та виконати дії, що зазначені у прийнятому пакеті, після чого надіслати відповідь та знову перейти в режим сну.
2. **Отримання запитів на отримання даних від головного модуля та вчасна відповідь на них.** Під час прийняття пакету сенсор перевіряє що прийшло в ньому, якщо прийшов код на отримання даних, сенсор збирає нові дані з датчиків, наприклад сенсор температури-вологості отримує нові дані про температуру та відносну вологість повітря. Після чого ці дані збираються у

пакет разом з деякими іншими даними, наприклад заряд батареї, рівень сигналу й інше, та надсилаються у головний модуль.

- 3. Отримання запитів на зміну даних від головного модуля та вчасне їх опрацювання.** Під час прийняття пакету сенсор перевіряє, що прийшло до нього, якщо прийшов код на зміну даних, сенсор виконує команди, які надійшли з головного модуля, наприклад виконавчий сенсор з реле, вмикає або вимикає певні реле згідно команд з головного модуля. Після чого зчитує й записує поточні дані з датчиків, ці дані збираються у пакет разом з деякими іншими даними, наприклад заряд батареї, рівень сигналу й інше, та надсилаються у головний модуль.
- 4. У разі різкої зміни параметрів або виходу їх за показник норми надіслати сигнал тривоги на головний модуль.** Мікроконтролер майже весь час знаходиться в режимі сну і, використовуючи таймер, виходить з нього та виконує процедуру збору даних з сенсорів, якщо показники різко зазнали зміни, для прикладу збільшення температури повітря вдвічі можливої свідчить про початок пожежі, або при збільшенні температури повітря вище 45°C це вже високий рівень температури повітря та з'являється можливість виходу показника за крайні рамки датчику температури (у датчика DHT11— це 50°C). Таким чином, при різкій зміні показника, або його високого/низького значення головний модуль буде аварійно проінформований.

Всі ці функції дозволяють оптимально працювати сенсорам та завжди повідомляти головний модуль у разі необхідності.

### **3.4 . Протокол зв'язку**

У роботі розроблено альтернативний протокол бездротового зв'язку. Програмно протокол реалізований на мові програмування C++ у середовищі Arduino IDE. Апаратно використовується радіомодуль NRF24L01+. В протоколі використовується топологія «зірка», схематичне зображення показано у Додатку М. Будова топології заключається в тому, що в центрі знаходиться головний модуль

котрий збирає дані з пристроїв системи, всі ж інші модулі системи спілкуються виключно з головним модулем, проте в разі проблем можуть використовувати інші модулі системи як міст до головного.

Протокол головного модуля має такі функції:

1. Постійно очікувати дані що можуть надійти, по аварійному каналі зв'язку — головний модуль має постійно працювати в режимі прийому даних, та постійно перевіряти канал на надходження даних через аварійний канал, що має негативний вплив на енергозбереження головного модуля, тому було вирішено відмовитися від використання апаратних пристосувань, а саме під'єднання ноги IRQ.
2. Пошук та реєстрація нових сенсорів — автоматично або за командою користувача здійснювати пошук нових сенсорів, при його знаходженні відбувається досить складний процес передачі даних, ускладнений він тим, що апаратна система перевірки доставки пакету не могла працювати з такими швидкостями роботи, постійною зміною адрес прийому. Етапи процесу реєстрації показано у Додатку Н.
3. Відправка запитів на отримання даних та подальше отримання відповіді від сенсора або виконавчого модуля — ця функція відбувається в кілька етапів:
  - 3.1. Відправка спеціального пакету для переведення сенсора або виконавчого пристрою у нормальний режим роботи з режиму низького енергоспоживання.
  - 3.2. Очікування відповіді від сенсора або виконавчого сенсора.
  - 3.3. Запис отриманих даних до оперативної пам'яті.
4. Відправка запитів на зміну даних, та подальше отримання відповіді від виконавчого модуля — ця функція відбувається в кілька етапів:
  - 4.1. Відправка спеціального пакету для переведення сенсора або виконавчого пристрою у нормальний режим роботи з режиму низького енергоспоживання.
  - 4.2. Відправка нової конфігурації виконавчого модуля (наприклад деякі контакти з реле увімкнути, інші вимкнути).

#### 4.3. Очікування пакету з результатом роботи (тобто в пакеті будуть надіслані дані про фактичну конфігурацію контактів).

Всі ці функції дозволяють оптимально працювати головному модулю та завжди отримувати свіжу інформацію, сигнали тривоги та надсилати нові конфігурації на виконавчі сенсори (Показано в Додатоку П). [12,13,14]

### 3.5. Тестування та аналіз результатів результати

Для тестування були розроблені та виготовлені дослідні зразки елементів системи «розумний дім». Для головного модулю системи була розроблена та виготовлена друкована плата. Для сенсору температури та вологості була також розроблена та виготовлена друкована плата, проте через те, що провідники були занадто тонкі під час нагрівання паяльником вони були пошкоджені, тому подальші дії з нею були припинені, згодом буде створена оновлена версія. А для виконавчого пристрою друкована плата перебуває у розробці. При подальшій роботі та перших тестуваннях системи виникли проблеми з пакетами підтвердження та радіусом роботи бездротових модулів. Оскільки розроблений протокол, динамічно змінює адресу надсилання запитів до сенсора або виконавчого пристрою, то сенсори циклічно не відповідали на запити з увімкненою апаратною системою підтвердження пакетів. Після вимкнення апаратної системи підтвердження та написання програмного аналогу ця проблема було усунено. Інша ж проблема вирішувалась тим, що паралельно до живлення радіомодуля потрібно було під'єднати конденсатори для згладжування стрибків живлення при вмиканні радіомодуля. Після під'єднання конденсатора були проведені експерименти на проникаючу здатність системи, результати котрих не задовільняли бажаних, тому що сигнал ловився лише через 1 стіну і було прийнято рішення програмно підняти потужність радіомодулів до 0 dBm(1 мВт). Після підняття потужності експеримент було повторено і модулі працювали відмінно, пробиваючи 3 бетонні стіни багатоквартирного житлового будинку. Щодо швидкості прийняття пакетів, то найвищий результат, який було отримано, склав близько 700 мікросекунд. Я вважаю, що це досить гарний результат.

Отже, ми отримали дослідний зразок системи «розумний дім» з конкурентно здатним протоколом бездротового зв'язку та з великою перспективою розвитку та удосконалення. У подальшому планується: введення голосового асистенту для спрощення керування системою, модернізація топології «зірка» та розробка нових модулів.

## ВИСНОВКИ

У роботі проаналізовано історію розвитку радіо, «розумного дому» та голосових асистентів, сучасні системи «розумного дому», такі як Ajax Systems, Broadlink, Sonoff, Xiaomi та протоколи передачі даних, а саме Z-Wave, ZigBee, Jeweller, й були виявлені основні переваги та недоліки сучасних рішень. Під час аналізу було вирішено створити альтернативний протокол зв'язку, який зможе конкурувати з аналогами, та буде мати відкритий код, що дозволить відносно маловартісним системам перейти на більш кращий протокол зв'язку вив вони використовують.

Розроблено бездротовий протокол зв'язку між елементами системи «розумний дім» на основі радіомодуля Nordic Semiconductor NRF24101+. Завдяки альтернативному протоколу зв'язку виготовлено дослідний зразок системи «розумний дім» на основі мікроконтролерів ATMEL ATMEGA328 та ESPRESSIF ESP8266-12E. Дані компоненти є загальнодоступними в Україні та мають незначну вартість. На базі мікроконтролера ESP8266-12E виконаний головний керуючий пристрій системи. За роботу сенсорів та виконавчих приладів відповідає ATMEGA328P на базі плати Arduino Nano.

Для керування системою «розумний дім» створено веб-інтерфейс, а у майбутньому буде створений мобільний додаток для операційної системи Android. Елементи голосового асистента майбутнього мобільного додатку для системи «розумний дім» будуть розроблені на основі сервісів Google.

На виконаному дослідному зразку окрім керуючого модуля використовуються сенсори температури та вологості та виконавчий пристрій з електромеханічні реле для керування електропостачанням побутових приладів. Дослідний зразок був протестований в реальних умовах двокімнатної квартири, в результаті проникла здатність скласти 3 бетонні стіни, а швидкість передачі даних 700 мікросекунд. Користувач системи отримував за потребою актуальні показники мікроклімату в приміщенні. Електромеханічні реле системи керують (вмикають та вимикають) освітленням, опаленням, водопостачанням, побутовими електроприладами тощо.

Користувач може переглядати дані від сенсорів та керувати роботою електромеханічних реле, як на персональному комп'ютері так і на мобільному пристрої, з будь-якою операційною системою, у веб-інтерфейсі.

Отже використання даного протоколу зв'язку дозволяє системі «розумного дому» працювати швидко, надійно, цей протокол зв'язку може бути альтернативним існуючим та конкурувати на ринку з іншими, як і більш дешевими так і більш дорогими рішеннями.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «ТОП 5: Рейтинг лучших систем «Умный дом» по производителям 2021 года». : [Електронний ресурс] – Режим доступу- <https://vencon.ua/articles/rejting-sistem-umnyu-dom-po-proizvoditelyam>
2. «Вікіпедія Розумний дім». [Електронний ресурс] – Режим доступу- [https://uk.wikipedia.org/wiki/Розумний\\_дім](https://uk.wikipedia.org/wiki/Розумний_дім)
3. «Мікроконтролери AVR»: [Електронний ресурс] – Режим доступу- <https://curlie.org/Computers/Hardware/Components/Processors/AVR/>
4. «Мікроконтролери PIC» : [Електронний ресурс] – Режим доступу- <https://www.microchip.com/>
5. «Мікроконтролери ESP8266» : [Електронний ресурс] – Режим доступу- [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)
6. «Голосовий асистент Siri» [Електронний ресурс] – Режим доступу- <https://uk.wikipedia.org/wiki/Siri>
7. «Голосовий асистент Google Assistant»: [Електронний ресурс] – Режим доступу- [https://uk.wikipedia.org/wiki/Google\\_Assistant/](https://uk.wikipedia.org/wiki/Google_Assistant/)
8. «Світ Радіо»: [Електронний ресурс] – Режим доступу- <http://proradio.org.ua/>
9. «Бездротовий протокол ZigBee»: [Електронний ресурс] – Режим доступу- <https://uk.wikipedia.org/wiki/ZigBee>
10. «Бездротовий протокол Z-wave»: [Електронний ресурс] – Режим доступу- <https://uk.wikipedia.org/wiki/Z-wave>
11. «История развития голосовых ассистентов и перспективы монетизации (на примере Алисы, Google Assistant, Siri, Alexa)»: [Електронний ресурс] – Режим доступу- <https://vernsky.ru/pubs/istoriya-razvitiya-golosovyh-assistentov-i-perspektivy-monetizatsii-na-primere-alisy-google-assistant-siri-alexa-5df8015cf2ad47219ee8ede4>
12. «Начало работы с ESP8266 NodeMcu v3 Lua с WiFi»: [Електронний ресурс] – Режим доступу- <https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/>

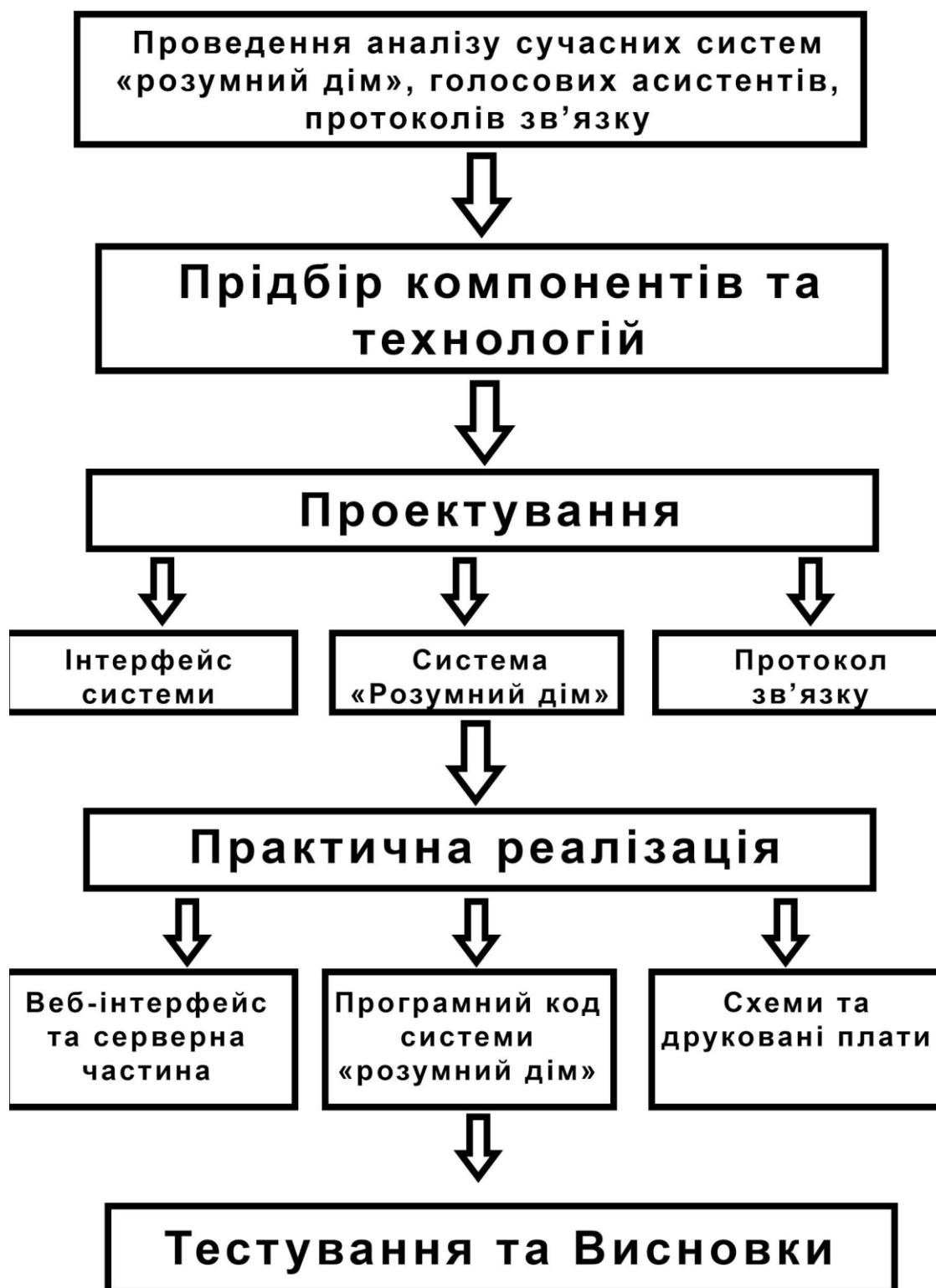
- 13.«Радио модуль NRF24L01+» :[Электронный ресурс] – Режим доступу-  
<https://wiki.iarduino.ru/page/NRF24L01-trema/>
- 14.«Соединяем две arduino по радиоканалу»: [Электронный ресурс] – Режим доступу-  
<https://lesson.iarduino.ru/page/urok-26-4-soedinyaem-dve-arduino-po-radiokanalu-cherez-nrf24l01/>
- 15.«Документация по RF24»: [Электронный ресурс] – Режим доступу-  
<http://maniacbug.github.io/RF24/classRF24.html>
- 16.«Настройка Arduino IDE для NodeMCU»: [Электронный ресурс] – Режим доступу-  
<https://robotclass.ru/articles/node-mcu-arduino-ide-setup/>
- 17.«FTP СЕРВЕР ESP8266»: [Электронный ресурс] – Режим доступу-  
<https://esp8266-arduinoide.ru/ftp/>
- 18.«Файловая система ESP8266»: [Электронный ресурс] – Режим доступу-  
[http://wikihandbk.com/wiki/ESP8266:Прошивки/Arduino/Работа\\_с\\_файловой\\_системой\\_в\\_аддоне\\_ESP8266\\_для\\_IDE\\_Arduino](http://wikihandbk.com/wiki/ESP8266:Прошивки/Arduino/Работа_с_файловой_системой_в_аддоне_ESP8266_для_IDE_Arduino)
- 19.«Сайт Ajax»: Офіційний сайт [Электронный ресурс] – Режим доступу-  
<https://ajax.systems/ua/>
- 20.«Сайт Broadlink»: Офіційний сайт [Электронный ресурс] – Режим доступу -  
<https://broadlink.com.ua/>
- 21.«Сайт Sonoff»: Офіційний сайт [Электронный ресурс] – Режим доступу -  
<https://sonoff.tech/>
- 22.«Конспект Хакера», [Электронный ресурс] – Режим доступу -  
<https://amperka.ru/product/hacker-workbook>
23. S. Fitzgerald, M. Shiloh «Arduino Projects Book»/ Torino, Italy, 2012. –175с.  
Режим доступу -  
[https://bastianvanhengel.files.wordpress.com/2016/06/arduino\\_projects\\_book.pdf](https://bastianvanhengel.files.wordpress.com/2016/06/arduino_projects_book.pdf)
- 24.Е.А. Тесля. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире»/ Санкт Петербург, 2008. – 195с.
- 25.Т. Р. Элсенпитер, Дж. Велт. «Умный Дом строим сами» / 2005. – 381с.

# ДОДАТКИ

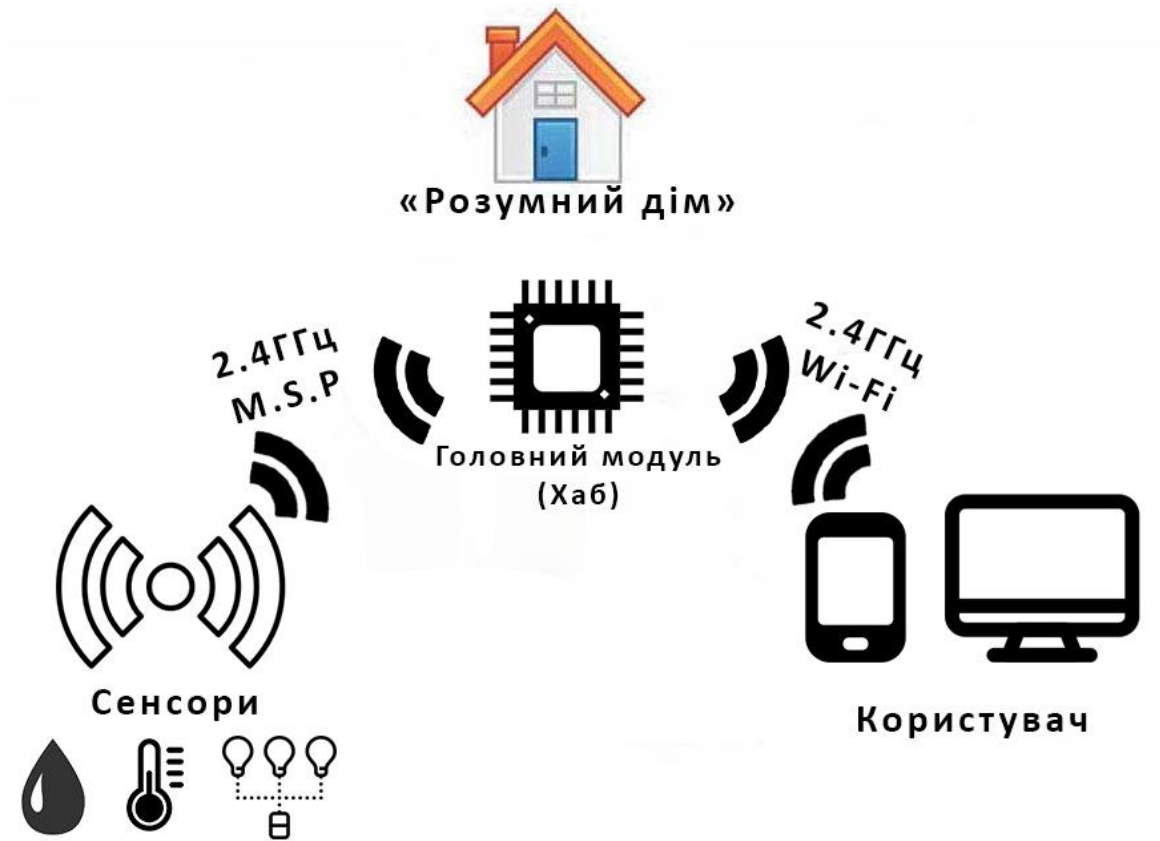
## Додаток А. Загальна схема «розумного дому»



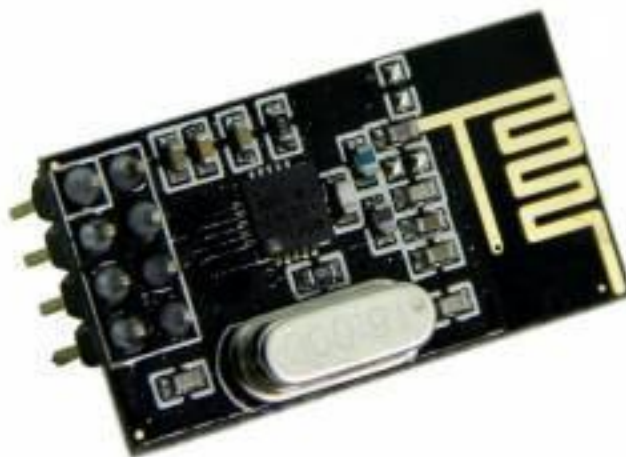
## Додаток Б. Схематичне зображення алгоритму розробки



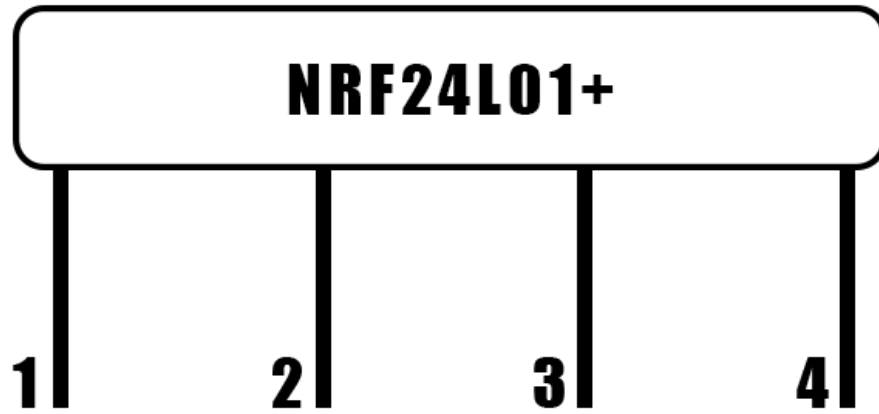
**Додаток В. Схематичне зображення  
альтернативної системи «розумний дім»**



**Додаток Г. Модуль бездротового радіозв'язку  
NRF24L01+ від Nordic Semiconductor**



Додаток Д. Схема розподілу каналів модуля

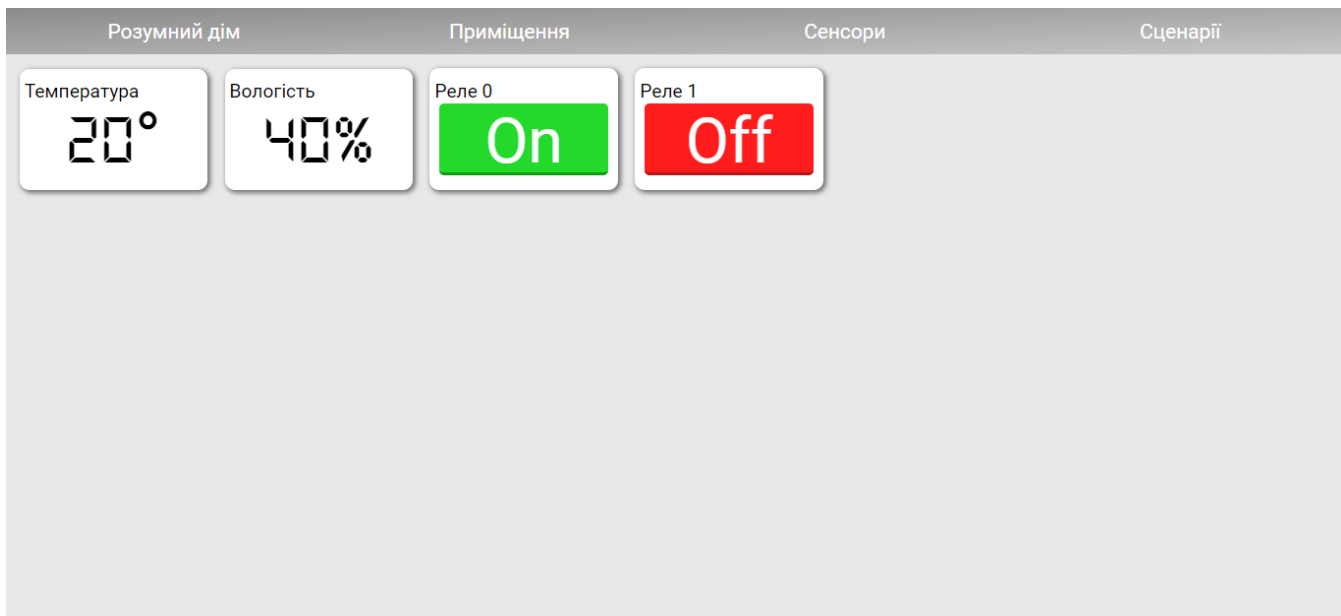


## Додаток Е. NRF24L01+ з підсилювачем сигналу та антеною





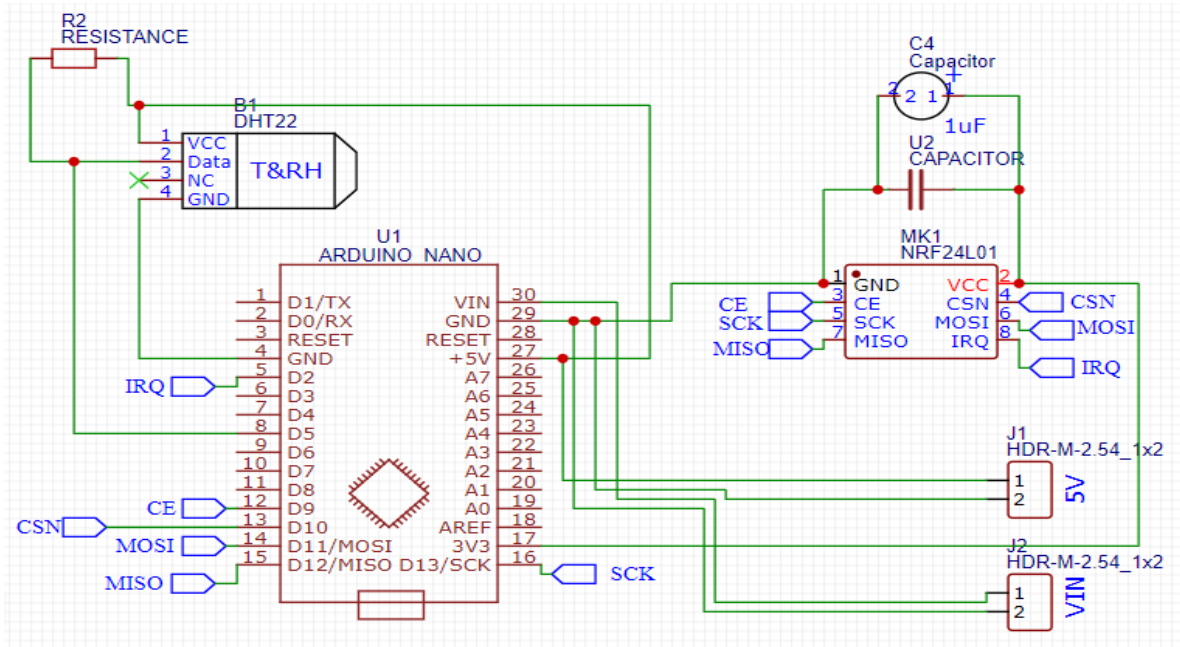
## Додаток Є. Веб-інтерфейс системи



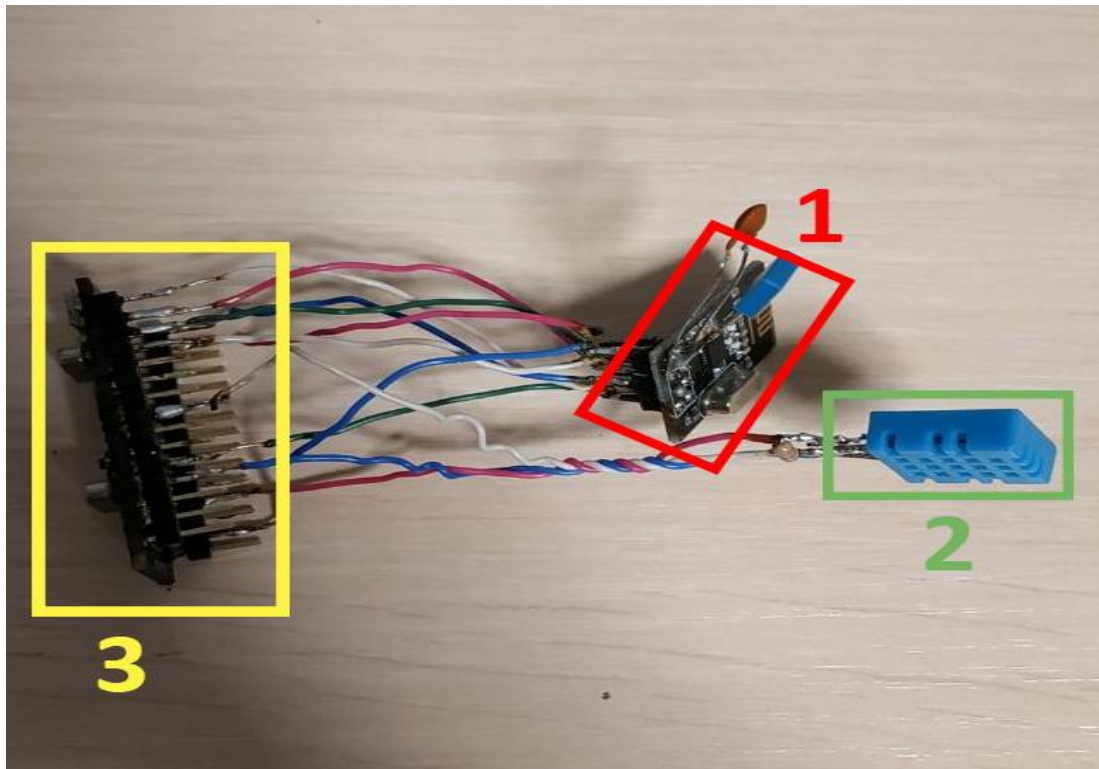
# Додаток Ж. Сенсор температури та вологості

## Додаток Ж.1

### Схема сенсора температури та вологості



### Готовий пристрій сенсора температури та вологості



1 — радіомодуль NRF24L01+

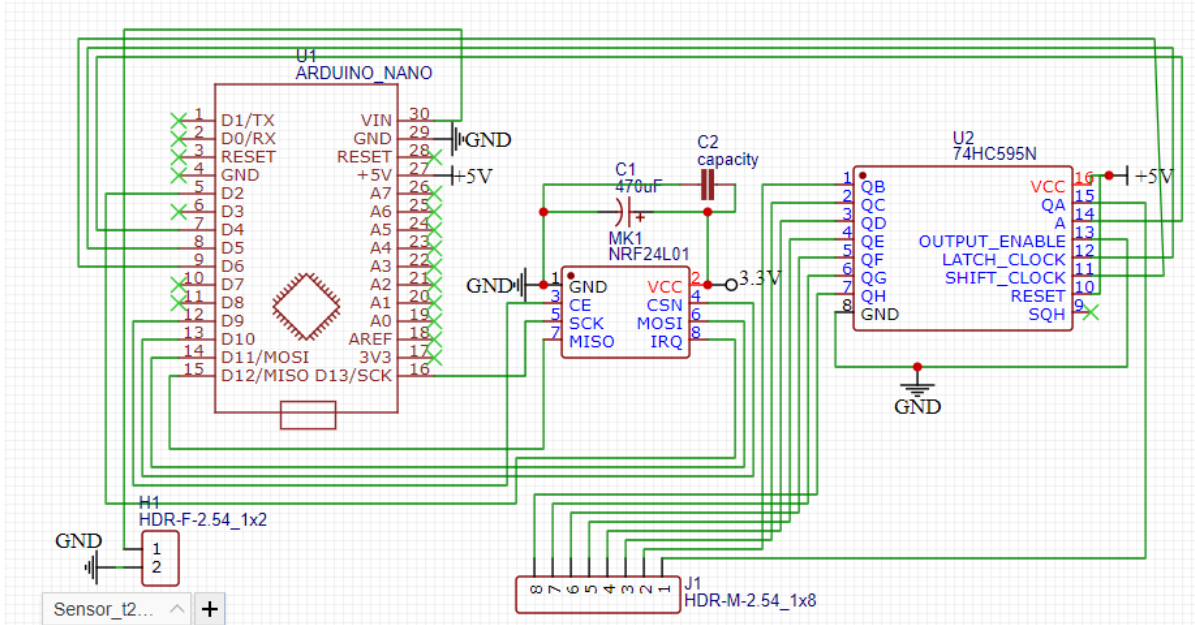
2 — датчик температури-вологи dht11

3 — плата Arduino Nano з мікроконтролером ATmega328p

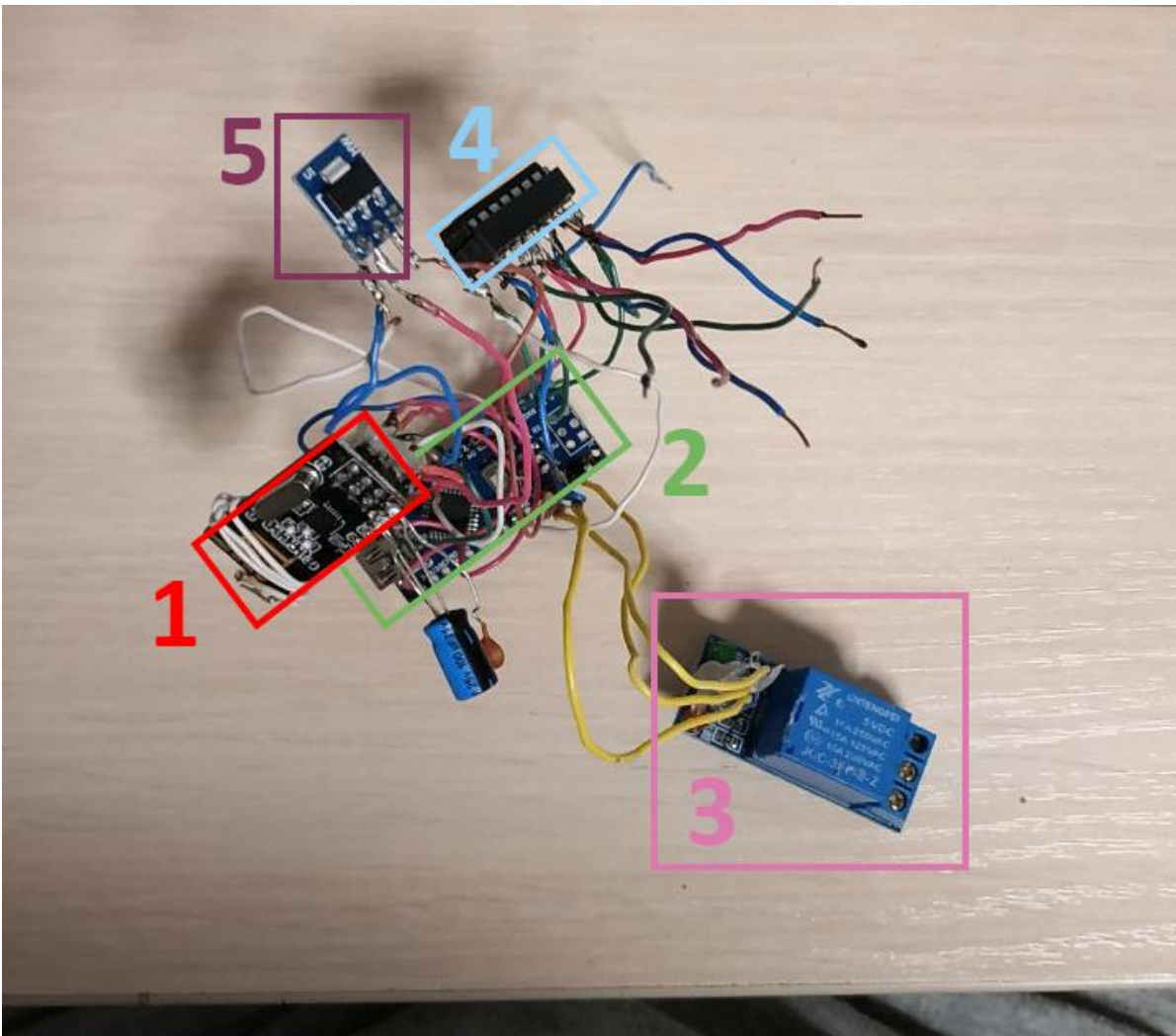
# Додаток К. Виконавчий сенсор

## Додаток К.1

### Схема виконавчого сенсора



### Готовий пристрій виконавчого сенсора

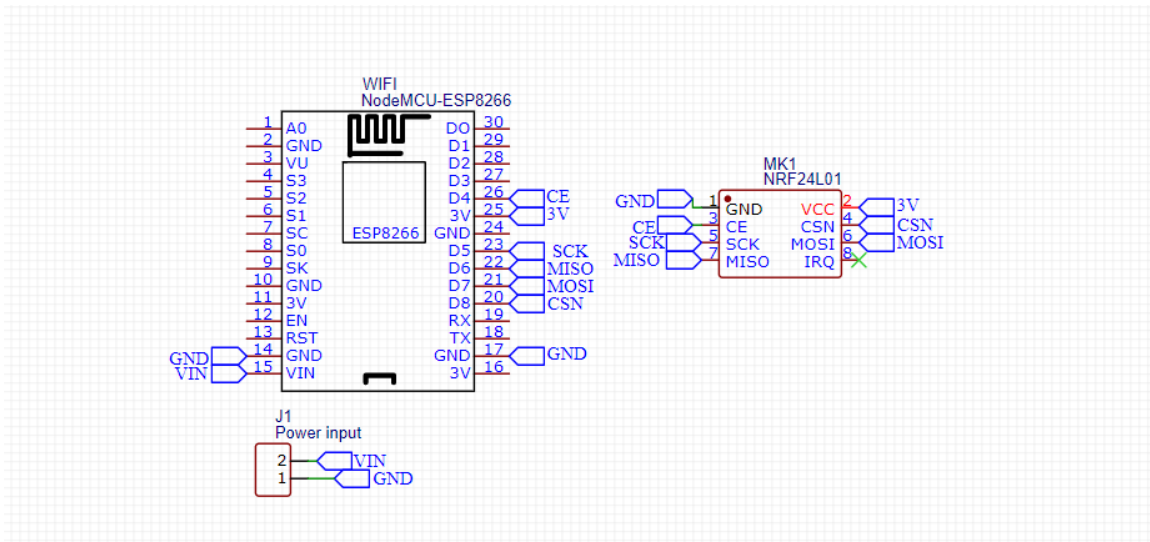


- 1 — радіомодуль NRF24L01+
- 2 — плата Arduino Nano з мікроконтролером ATmega328p
- 3 — електро-механічне реле
- 4 — здвиговий регістр 74HC595
- 5 — стабілізатор напруги з 5В до 3.3В

## Додаток Л. Головний модуль системи

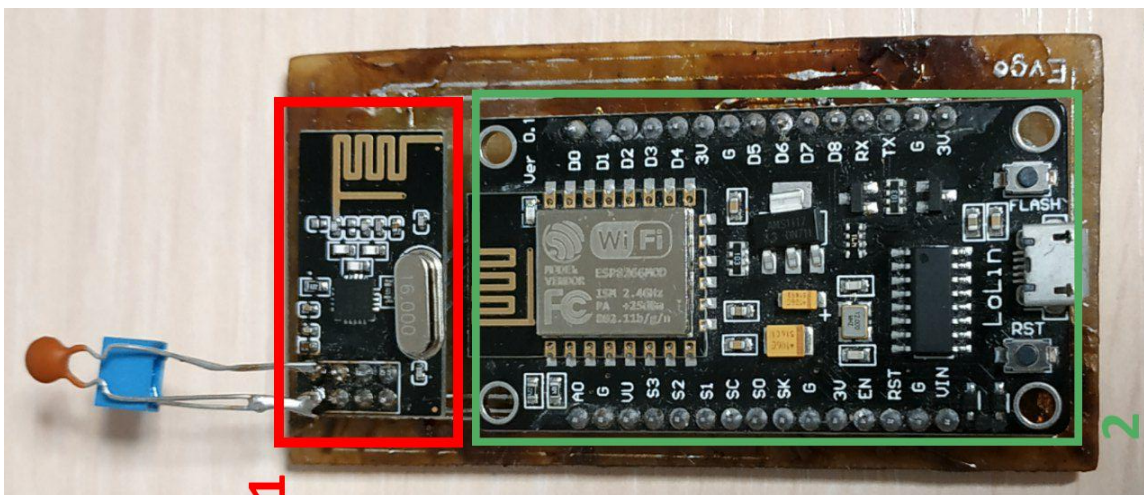
### Додаток Л.1

#### Схема головного модулю системи



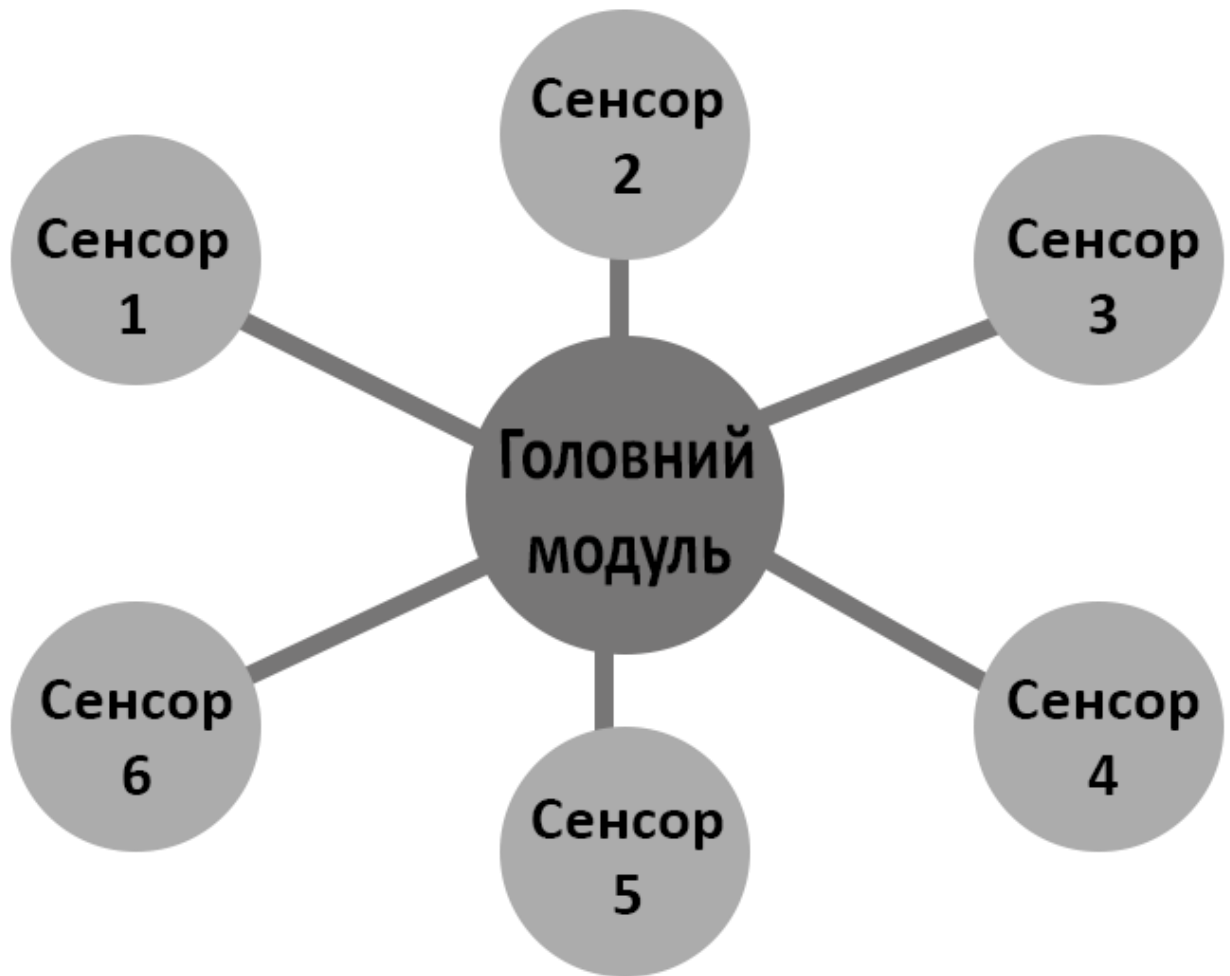
### Додаток Л.2

#### Готовий пристрій головного модулю системи



1 — радіомодуль NRF24L01+

2 — плата NodeMCU v3 з мікроконтролером esp8266 на борту

**Додаток М. Схема топології «зірка»**

## Додаток Н. Процес реєстрації модулів





## Додаток П. Програмний код

### Головні функції головного модуля:

#### Функція реєстрації нового модуля

```

1 boolean register_() {
2   Serial.println("Reg function started!");
3   radio.stopListening(); //Вимкнення прослуховування радіо ефіру
4   Serial.print("[STEP-1] Scanning:");
5   radio.openWritingPipe(address[REG_ADDR]); //Відкриваємо канал
6   //передачі даних на адресу реєстрації нових модулів
7   uint8_t status_ = 200; //200 - код на дозвіл реєстрації!
8   radio.write(&status_, sizeof(status_)); //Відправка дозволу
9   radio.startListening(); //Увімкнення прослуховування радіо ефіру
10  uint8_t i = 0;
11  while (i < 7) {
12    if (radio.available()) { //Перевіряємо чи є нові прийняті пакети
13      break;
14    }
15    delay(20);
16    i++;
17  }
18  if (i == 7) {
19    Serial.println("Not found!"); //Якщо не отримали функція завершена
20    return false;
21  }
22  Serial.println("Found!");
23  Serial.print("[STEP-2] Verifying:");
24  radio.read(&reg_pack, sizeof(reg_pack));
25  //Читаємо пакет (апаратна перевірка автоматична) та перевіряємо програмно
26  if (reg_pack.id_sensor != 0 or reg_pack.signal_lvl < 100) {
27    //Verifying...
28    Serial.println("Sensor not correct");
29    Serial.println("[ERROR] REG ERR");
30    //Апаратна перевірка пройдена, проте
31    //програмну перевірку пакет не пройшов, тобто
32    //пакет був не коректно надісланий
33    return false;
34  }
35  Serial.println("Sensor OK");
36  reg_pack.id_sensor = count_of_sensors() + 1; //Калькуляція нового id
37  sensors_template[0] = count_of_sensors() + 1; //Внесення даних до RAM
38  memcpy(reg_pack.person_pipe[0], sensors_template, sizeof(reg_pack.person_pipe[0]));
39  radio.stopListening();
40  //Вимикаємо прослуховування та відправляємо новий пакет для сенсора
41  radio.write(&reg_pack, sizeof(reg_pack));
42  radio.startListening();
43  i = 0;
44  //Чекаємо відповіді
45  while (i < 7) {
46    if (radio.available()) {
47      break;
48    }
49    delay(30);
50    i++;
51  }
52  if (i == 7) {
53    Serial.println("[ERROR] REG ERR, not recived confirm packet!");
54    return false; //Фінальна відповідь не отримана
55  }
56  Serial.print("[STEP-3] Checking register:");
57  uint8_t sensor_id = 0;
58  radio.read(&sensor_id, sizeof(sensor_id));
59  if (sensor_id != reg_pack.id_sensor) {
60    Serial.print("Recived id, not correct");
61    Serial.println("[ERROR] REG ERR");
62    return false; //Перевірка фінального пакета не пройдена
63    //Реєстрація - відмінена
64  }
65  Serial.print("Sensor confirm!");
66  Serial.println("");
67  sensor_reg(reg_pack.id_sensor, reg_pack.type_sensor);
68  //Запускаємо функцію запису даних до файлу конфігурації
69  return true;
70 }

```

## Функція запису нового пристрою у файлову систему

```
1 boolean sensor_reg(uint8_t id, uint8_t type) {
2   Serial.print("Reg sensor:");
3   File conf_file = SPIFFS.open("/config.json", "r");
4   //Відкриваємо файл(з файлової системи)
5   //конфігурації на читання
6   if (!conf_file) {
7     Serial.println("file open failed");
8     // "Відкрити не вдалось"
9   }
10  DynamicJsonDocument doc(Memory_json);
11  //Виділяємо динамічно пам'ять на читання файлу
12  deserializeJson(doc, conf_file);
13  //"Дешифрували" - зробили JSON зрозумілим для нас
14  conf_file.close();
15  //Закрили раніше відкритий файл
16  conf_file = SPIFFS.open("/config.json", "w");
17  //Зновуш таки його відкрили проте з дозволом на запис
18  delay(5);
19  JSONArray Sensor =
20  doc["Sensors"].createNestedArray(String(id));
21  //Створюємо об'єкт нового сенсора для
22  //подальшого запису у файл
23  Sensor.add(type);
24  Sensor.add("Empty");
25  //Конфігуруємо поля для коректної роботи
26  if (serializeJson(doc, conf_file) == 0) {
27    //"Шифруєм" дані у формат JSON і записуєм у файл
28    //Перевіряємо чи вдалося записати щось у файл
29    conf_file.close();
30    Serial.println("ERR");
31    return false;
32    //Закриваємо файл і вертаєм значення помилки
33  }
34  conf_file.close();
35  Serial.println("OK");
36  return true;
37  //Закриваємо файл і вертаєм значення успіху
38 }
```

## Менеджер запису даних у головний модуль (для сенсора температури і вологості)

```

1 uint8_t data_manager_write(type_one_sensor & pack) {
2     /*
3     Приймаємо: пакет з прийнятою інформацією
4     Вертаємо:
5     1 - Пам'ять знайдена, дані записані
6     2 - Пам'ять знайдена, дані не записані
7     3 - Пам'ять не знайдена, Фатальна помилка
8     4 - Помилка дані не коректно подані
9     */
10    boolean founded = false;
11    //Флаг знайденої пам'яті
12    uint8_t found_index = 0;
13    //Індекс знайденої пам'яті
14    Serial.println("[INFO] Manager started for 1 type");
15    Serial.print("Sensor id: ");
16    Serial.print(pack.id);
17    if (pack.type == 0) {
18        Serial.println("Type ERROR type is 0");
19        return 4;
20        //Помилка передачі даних
21    }
22    uint8_t memory = sizeof(last_type_one) /
sizeof(last_type_one[0]);
23    //Кількість зареєстрованих сенсорів типу один
24    for (uint8_t i = 0; i < memory; i++) {
25        if (last_type_one[i].id == pack.id) {
26            founded = true;
27            found_index = i;
28            break;
29        }
30    }
31    //Минулі дані знайдені, нові дані будуть
32    //перезаписані у цю комірку
33    if (!founded) {
34        //Минулі дані не знайдено, шукаємо порожню комірку
35        Serial.println("Previous data from this sensor not
found");
36        Serial.println("Searching empty memory");
37        for (uint8_t i = 0; i < memory; i++) {
38            if (last_type_one[i].id == 0) {
39                founded = true;
40                found_index = i;
41                break;
42            }
43        }
44    }
45    if (!founded) {
46        //Фатальна помилка, пам'ять заповнена
47        Serial.println("Empty memory not found!");
48        return 3;
49    }
50    Serial.println("Ok, memory founded!");
51    last_type_one[found_index].id = pack.id;
52    last_type_one[found_index].type = pack.type;
53    last_type_one[found_index].v_bat = pack.v_bat;
54    last_type_one[found_index].status_sensor =
pack.status_sensor;
55    last_type_one[found_index].signal_lvl = pack.signal_lvl;
56    last_type_one[found_index].temp = pack.temp;
57    last_type_one[found_index].humdacity = pack.humdacity;
58    last_type_one[found_index].heat_index = pack.heat_index;
59    last_type_one[found_index].kef = pack.kef;
60    //Заповнення масиву новими даними
61    Serial.println("Verifying..");
62    if (last_type_one[found_index].id == pack.id &&
last_type_one[found_index].temp == pack.temp &&
last_type_one[found_index].humdacity == pack.humdacity) {
63        //Перевірка, записаних даних
64        Serial.println("Verified!");
65        return 1;
66        //Перевірка пройдена, вертаємо успіх
67    }
68    return 2;
69    //Перевірка не пройдена, вертаєм помилку
70 }

```

## Головні функції у сенсорі температури та вологості

### Опитування датчику вологи та температури

```
1  if (millis() - timing > 1000) {
2    timing = millis();
3    uint8_t temp_t = uint8_t(dht.readTemperature());
4    uint8_t temp_h = uint8_t(dht.readHumidity());
5    float temp_i = dht.computeHeatIndex(last_temp, last_humd,
    false);
6    //Збираємо данні з сенсорів
7    if ((float)temp_t / last_temp > 1.2 or (float)temp_t /
    last_temp < 0.9) {
8      //Пізка зміна параметрів температури
9      Serial.println("PALUNDRA TEMPERATURE CHANGE!!!");
10     uint8_t mode = 1;
11     palundra(mode, temp_t, last_temp);
12   }
13   if ((float)temp_h / last_humd > 1.2 or (float)temp_h /
    last_humd < 0.5) {
14     //Пізка зміна параметрів відносної вологості повітря
15     Serial.println("PALUNDRA HUMDACITY CHANGE");
16     uint8_t mode = 2;
17     palundra(mode, temp_h, last_humd);
18   }
19   last_temp = uint8_t(temp_t);
20   last_humd = uint8_t(temp_h);
21   last_heat_index = temp_i;
22   //Записуємо нові дані у оперативну пам'ять
23 }
```

## Функція тривоги

```
1 boolean palundra(uint8_t& mode, uint8_t& now, uint8_t&
  pred) {
2   alert.id = id_sensor;
3   alert.data_type = mode;
4   //Заповнення пакету даних для тривог
5   Serial.println("Palundra mode started!");
6   Serial.print("Mode:");
7   Serial.println(mode);
8   Serial.print("Now:");
9   Serial.println(now);
10  Serial.print("Previous data:");
11  Serial.println(pred);
12  //Вивід даних в консоль
13  if (now > pred) {
14    Serial.println("Very fast upping info");
15    alert.updown = 1;
16  }
17  else {
18    Serial.println("Very fast falling info");
19    alert.updown = 0;
20  }
21  //Визначення чи падає, чи зростає показник
22  radio.stopListening();
23  //Вимикаємо прийом, вмикаємо передачу
24  radio.openWritingPipe(address[1]);
25  //Переключаємось на канал тривог
26  radio.write(&alert, sizeof(alert));
27  //Відсилаємо пакет
28  radio.startListening();
29  return true;
30  //Переводимо систему у звичний режим
31  //Прослуховування, та вертаємо успіх
32 }
```

## Головні функції виконавчого пристрою

### Зміна статусу контакта

```
1 void write_register_arr(boolean pins[8]) {
2   chanel = 0b00000000;
3   //8 каналів реле представляємо в виді бітів в байті
4   for (uint8_t i = 0; i < 8; i++) {
5     bitWrite(chanel, i, !pins[i]);
6     //Якщо реле низького логічного рівня
7   }
8   digitalWrite(latchPin, LOW);
9   //Вимикаємо контакт захисту запису на 74HC595
10  shiftOut(dataPin, clockPin, MSBFIRST, chanel);
11  //Передача інформації на 74HC595
12  digitalWrite(latchPin, HIGH);
13  //Вмикаємо контакт захисту запису на 74HC595
14 }
```

## Частина коду, котрий регулює відповіді головному модулю

### виконання його команд

```
1  Serial.println("Request on change data waiting a data...");
2  uint8_t i = 0;
3  while (i < 15) {
4      if (radio.available()) {
5          break;
6      }
7      i++;
8      delay(20);
9  }
10 if (i < 15) {
11     Serial.println("Data recived!");
12     radio.read(&data_recived, sizeof(data_recived));
13     //Читаємо пакет котрий прийшов
14     Serial.println("Veryfing...");
15     if (data_recived.id == id_sensor) {
16         //Наш пакет ? Якщо так то продовжуємо
17         Serial.println("Veryfied!");
18         Serial.println("Working....");
19         write_register_arr(data_recived.relays);
20         //Виконуємо команди з головного модуля
21         Serial.println("Pins changed!");
22         Serial.println("Send a report..");
23         //Готуємо дані для головного модуля
24         for (uint8_t i = 0; i < 8; i++) {
25             data_recived.relays[i] = !bitRead(channels, i);
26             Serial.print("Chanell ");
27             Serial.print(i);
28             Serial.print(": ");
29             Serial.println(!bitRead(channels, i));
30         }
31         radio.stopListening();
32         //Вимикаємо прослуховування
33         radio.openWritingPipe(address[3]);
34         //Встановлюємо канал на головний модуль
35         radio.write(&data_recived, sizeof(data_recived));
36         //Відсилаємо дані, та вертаєм режим прослуховування
37         radio.startListening();
38         Serial.println("Data sended!");
```