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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN”

КбПЗ_2025

Виконав здобувач вищої освіти
IV курсу, групи КІ-21-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Удудяк Б.О.
«____» _____ 2025 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Петренюк В.І.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
Галузь знань ._12 _____*"*Інформаційні технології*"*_____
Спеціальність *123 "*Комп'ютерна інженерія*"*
Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерна інженерія*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 17 » січня 2025 року

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

Удудяку Богдану Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN*

2. Керівник роботи *Петренюк Володимир Ілліч,канд. фіз.-мат. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 47-02 від 17.01.2025 року

3. Строк подання студентом роботи до захисту *23.05.2025 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою роботи є розробка програмного забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN*

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

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

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

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

КБПЗ-2025

7. Дата видачі завдання « 17 » січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2025 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2025 р.	
3.	Розробка моделі компонента	20.03.2025 р.	
4.	Розробка структур даних	25.03.2025 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2025 р.	
6.	Програмування алгоритмів	10.04.2025 р.	
7.	Оформлення ПЗ	17.04.2025 р.	
8.	Попередній захист роботи	23.05.2025 р.	

КбПЗ_2025

Дата видачі завдання
« 17 » січня 2025 р.

Підпис керівника
_____ Петренюк В.І.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2025 р.

Підпис здобувача
_____ Удудяк Б.О.
(прізвище та ініціали)

АНОТАЦІЯ

Удудяк Б.О. Програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи IoT інтелектуального будинку з використанням мережі LPWAN.

Метою розробки є програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN.

Результат роботи – програмна реалізація системи IoT інтелектуального будинку з використанням мережі LPWAN.

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

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

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

Програму розроблено в середовищі Builder C++.

Ключові слова: комп'ютерна інженерія, IoT, інтелектуальний будинок, LPWAN

КПЗ – 2025

ABSTRACT

Ududyak B.O. Software for the IoT system of an intelligent house using the LPWAN network. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for the IoT system of an intelligent house using the LPWAN network.

The purpose of the development is the software for the IoT system of an intelligent house using the LPWAN network.

The result of the work is the software implementation of the IoT system of an intelligent house using the LPWAN network.

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

A convenient user interface has been developed. Instructions for working with software are provided.

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

The program was developed in the Builder C++ environment.

Keywords: computer engineering, IoT, smart home, LPWAN

КОПЗ_2025

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 2
ВСТУП..... 3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 5
1.1 Призначення системи..... 5
1.2 Область застосування..... 5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... 7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 24
2.3 Розгорнута постановка завдання 26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 28
3.1 Опис функціонування системи 28
3.2 Розробка структурної схеми..... 30
3.3 Розробка функціональної схеми 34
3.4 Розробка діаграми процесів..... 37
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 39
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 39
4.2 Захист розробленого програмного забезпечення..... 55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 61
6 ОСНОВНІ ВИСНОВКИ..... 68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 70

КОПЗ – 2025

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АРМ	–	автоматизоване робоче місце
АСУ	–	автоматизована система управління
ДБЖ	–	джерело безперебійного живлення
ДКС	–	домашня кабельна мережа
ДУ	–	дистанційне управління
ЕОМ	–	електронно-обчислювальна машина
ЕФ	–	екранна форма
ЗТО	–	звукова трансляція й оповіщення
ІБ	–	інтелектуальний будинок
ІЧ	–	інфрачервоний
ОВК	–	управління опаленням, вентиляцією й кондиціюванням
ОДС	–	оперативна диспетчерська система
ОПС	–	охоронно-пожежна сигналізація
ПДУ	–	пульти дистанційного управління
ПЗ	–	програмне забезпечення
ПЛК	–	програмувальні логічні контролери
ПМО	–	програмно-математичного забезпечення
РК	–	рідкокристалічний
СКК	–	система кабельних комунікацій
ТЗ	–	технічне завдання
EIB	–	європейська інсталяційна шина
LPWAN	–	технологія інтелектуального дому

КОПЗ-2025

ВСТУП

Актуальність теми. Рішення для телеметрії й телемеханіки застосовуються вже більше сторіччя, але з появою модного терміна «Інтернет речей» потік інвестицій у дану область істотно збільшився. Один з нових напрямків – територіально розподілені мережі LPWAN, що забезпечують бездротове підключення пристроїв з дуже економним режимом енергоспоживання: автономна робота від акумуляторів до 10 років. Впровадження LPWAN відкриває безліч перспектив для всіх гравців ринку бездротових мереж передачі даних.

Існує безліч прогнозів щодо росту числа підключених «речей», причому цифри, що вказуються в них, можуть відрізнятися на порядок. Так, скажемо, експерти Gartner думають, що до 2027 року буде налічуватися «усього» близько 21 млрд підключених пристроїв, а в Intel виходять із того, що їх буде в десять разів більше – 200 млрд.

Через такі розбіжності, у якості «відправної точки» найкраще взяти досить стриманий прогноз Ericsson Mobility Report. Відповідно до цього документа, загальне число підключених пристроїв у період з 2021 по 2025 рік виросте приблизно у два рази: з 15 до 28 млрд штук, – причому в ньому враховуються й традиційні засоби для зв'язку«людина – людина» (мобільні й фіксовані телефони), а також для роботи в Мережі (ПК, ноутбуки, планшети).

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN.

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

– Огляд існуючих систем IoT інтелектуального будинку з використанням мережі LPWAN.

КОПІЯ - 2025

– Дослідження системи IoT інтелектуального будинку з використанням мережі LPWAN.

– Програмна реалізація системи IoT інтелектуального будинку з використанням мережі LPWAN.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі IoT інтелектуального будинку з використанням мережі LPWAN.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КбПЗ_2025

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

По даним J'son & Partners Consulting, «загальне число підключених пристроїв у розподілених системах телеметрії Україні» за розглянутий період також збільшиться приблизно у два рази: з 16,1 до 32,6 млн штук. Хоча аналітики Ericsson і J'son & Partners використовували різні критерії для оцінки, навіть по грубій прикидці видно, що український ринок IoT становить істотно менш 1% від загальносвітового. Цікаво відзначити, що, згідно Ericsson Mobility Report, в 2021 році загальне число традиційних комунікаційних пристроїв (10,1 млрд штук) більш ніж у два рази перевершувало число підключених «речей» IoT (4,6 млрд штук). В 2025 році розклад сил зміниться вже на користь IoT: 15,7 проти 11,8 млрд пристроїв. При цьому число традиційних засобів зв'язку буде збільшуватися лише на кілька відсотків у рік, тоді як число підключених речей – більш ніж на 20%.

У дослідженні Ericsson Mobility Report окремо враховуються пристрої IoT, підключені через мережі стільникового зв'язка (cellular IoT) і за допомогою інших технологій бездротового зв'язку (non-cellular IoT). Останніх в 2021 році було приблизно в 10 разів більше. Подібне співвідношення збережеться й в 2025 році.

КПЗ_2025

1.2 Область застосування

Областю застосування є інтелектуальний будинок. На сьогоднішній період не існує строгого визначення поняття «розумний будинок» або «інтелектуальний будинок». Поки що кожний користувач самостійно визначає основні функції й інженерні системи у квартирі й будинку, на яких можуть бути реалізовані ці функції шляхом додаванням інтелекту керування.

І кожна така система підвищує рівень інтелектуальності будинку надаючи новий рівень безпеки, функціональності й комфорту. Головна роль у реалізації цих проектів належить компаніям, що здійснюють проектування й установку системи розумний будинок, при спільній участі власника або хазяїна зазначеного будинку.

Система розумний будинок включає, але не обмежує, що випливають функції: Дистанційний контроль роботи інженерних систем будинку – опалення, електропостачання, газової служби, водопостачання й каналізації, охоронної системи. А також систему дистанційного керування.

Оповіщення й зв'язки: наприклад – споживач може шляхом дзвінка на номер системи розумного будинку одержати інформацію (усну або по СМС) стани електроспоживання, опалення), про включених споживачів (наприклад – праска, телевізор, освітлення, газоспоживання й роботі інших систем).

Двосторонній зв'язок і контроль через системи зв'язку основних показників роботи інженерних систем.

Автоматична інформація споживача про аварійну ситуацію, з рекомендаціями з усунення ситуації, і з автоматичним застосуванням деяких функцій по ліквідації аварійних ситуацій шляхом передачі команд через СМС або за допомогою номеронабирача телефонного апарата.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КОПЗ – 2025

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Сьогодні існує розумна версія майже кожного домашнього пристрою, який тільки можна придумати. Ці продукти (зазвичай) підключаються до Інтернету, тож ви можете керувати ними зі свого телефону через супутню програму. Більшість також працює з різними формами голосового керування. Нижче наведено короткий перелік найкращих інтелектуальних пристроїв, які ми протестували для кожної зони будинку. Продуктів так багато, що ми розмістили їх по кімнатах, щоб усе було впорядковано, але це не означає, що більшість із цих гаджетів не можна використовувати будь-де. Перевірте незабаром, оскільки ми регулярно тестуємо нові продукти та додаємо їх до цього посібника.

Передній двір: створення добре освітленого та безпечного входу

У ідеальному розумному домі ви знайомитеся з технологіями ще до того, як ступите через вхідні двері. Деякі з найкорисніших підключених домашніх гаджетів знаходяться у передньому дворі, починаючи з зовнішніх камер спостереження. TP-Link Tapo C120 – наш найкращий вибір для стеження за домом із телефону. Ця стійка до погодних умов камера безпеки з високою роздільною здатністю не тільки знімає відео, викликане рухом, але й надсилає безкоштовні інтелектуальні сповіщення, які відрізняють людей, домашніх тварин, транспортні засоби та інші події руху. Для моделі з 2K-відео, кольоровим нічним баченням і вбудованими прожекторами ви не зможете побити її ціну.

КПЗ-2025

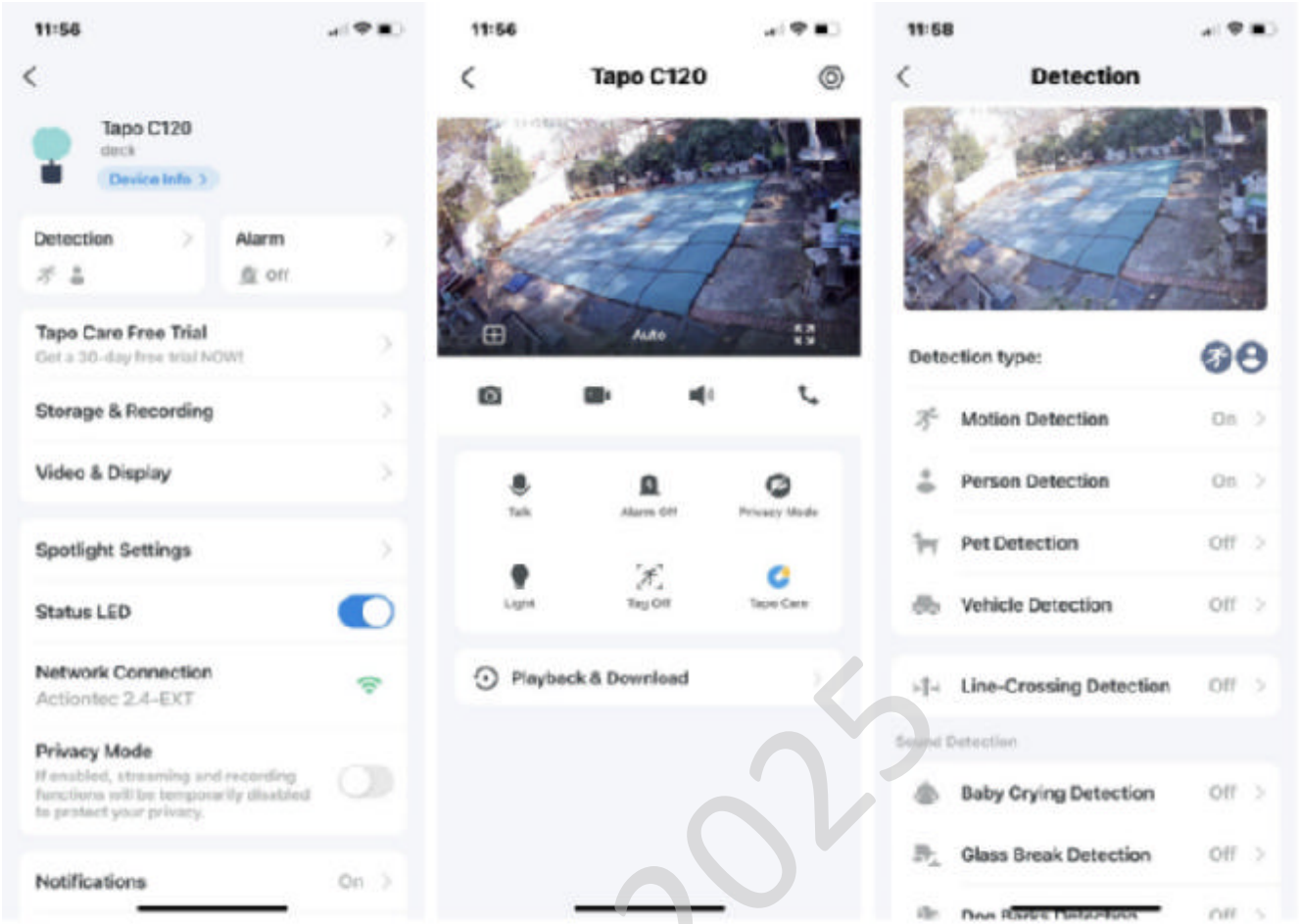

Рисунок 2.1 – Скріншоти з додатка-супутника Tapo C120

Якщо вам захочеться перевірити, чи стукає у ваші двері кур'єр чи небажаний адвокат, **відеодзвінок** дозволить вам побачити, хто там, не встаючи з дивана. Окрім цієї зручності, дверні відеодзвінки можуть служити першою лінією захисту від крадіжки майна, вторгнення в житло та піратів у під'їздах. Вони не тільки дозволяють вам побачити та поговорити з людиною на вулиці, але вони також записують кадри відвідувачів, які підходять до ваших дверей, коли ви відсутні або не можете відповісти. Наша улюблена модель, відеодверний дзвінок Arlo, забезпечує чітке відео у форматі HD або 2K із широким полем огляду по діагоналі 180 градусів, дає можливість дротового чи бездротового налаштування, працює з багатьма пристроями сторонніх виробників і підтримує голосове керування.

КБПЗ_2025

Ще одна технологія, яка вам може знадобитися на ваших вхідних дверях, – це **розумний замок**. Ці пристрої, мабуть, найважливіша частина по-справжньому підключеного дому, забезпечують безпеку та зручність, дозволяючи вам приходити та виходити, коли вам заманеться, і стежити за тим, хто входить і виходить із вашого дому, коли вас немає. Наш поточний найкращий вибір – це Ultraloq Bolt Fingerprint, який додає підтримку HomeKit до нашого попереднього фавориту, залишаючись при цьому надійним і універсальним розумним замком, який дозволяє отримати доступ або замкнути за допомогою голосу та відбитка пальця, а також за допомогою клавіатури, мобільного додатка чи традиційних ключів.

Якщо ви бажаєте усамітнитися, але все ж хочете впустити сонце, Smart Glass Feather River Doors пропонує найкраще з обох світів. Ці розумні вхідні двері перемикаються між непрозорим і прозорим склом натисканням кнопки або голосовою командою.

Щоб отримати найвищий рівень захисту, розгляньте **систему безпеки розумного будинку** з професійним моніторингом. Один із наших найкращих варіантів, ADT Command, – це значні інвестиції, але він пропонує практично все, що вам потрібно для максимального захисту. Порівняно з Vivint, іншим переможцем ADT трохи дорожчий, але пропонує чудове обслуговування клієнтів.

У якості більш доступної альтернативи « **зроби сам» розумні системи безпеки будинку** ідеально підходять власникам будинків, які хочуть заощадити гроші, самостійно стежачи за своїми домівками. Система домашньої безпеки SimpliSafe, наш улюблений варіант «зроби сам», дає змогу захистити та стежити за своїм домом здалеку, не витрачаючи багато коштів і не підписуючи довгостроковий контракт.

КБПЗ-2025

Рисунок 2.2 – Система домашньої безпеки SimpliSafe

Вітальня: Затишні зручності для розваг і відпочинку

Коли справа доходить до технологій для вітальні, телевізори та звукові панелі є очевидними зірками шоу. Але багато інших гаджетів спрямовані на підвищення атмосфери, зручності та безпеки вашого основного житлового простору.

Для початку **розумна колонка** є обов'язковою умовою для вітальні. Окрім відтворення музики, наші улюблені моделі служать своєрідним центральним командним центром, що дозволяє керувати практично будь-яким іншим підключеним домашнім гаджетом голосом. Першим кроком у виборі є рішення про те, яка платформа голосової допомоги підходить для вашого дому: Alexa, Google Assistant або Siri. Усі три підтримують більшість основних брендів пристроїв для розумного дому.

Якщо Alexa є вашим голосовим помічником, ми пропонуємо Echo Studio, який пропонує справді вражаючу якість звуку за свою ціну, або Amazon Echo

КБПЗ_2025

четвертого покоління як більш доступну альтернативу. Шанувальники Google Assistant можуть використовувати Nest Audio. А якщо ви віддаєте перевагу Siri, ви можете придбати HomePod або HomePod mini. Якщо ви підете цим шляхом, просто переконайтеся, що будь-які інші розумні домашні пристрої, які ви купуєте, сумісні з Apple HomeKit або Matter. Тим часом Sonos Era 300 і 100 підтримують Alexa, але не підтримують Google Assistant або Siri.

Як альтернативу або на додаток до розумної колонки вам може знадобитися **розумний дисплей**. Вони пропонують такі ж функції голосового помічника, як і розумний динамік, але з корисним сенсорним екраном, на якому ви можете переглядати прямі трансляції з інтелектуального дверного дзвінка або камери безпеки, здійснювати відеодзвінки та керувати підключеними гаджетами кількома дотиками. Коли ваш розумний дисплей у вітальні не використовується, він може служити цифровою фоторамкою для відображення ваших особистих знімків або підібраного мистецтва. Amazon Echo Show 10 з підтримкою Alexa та Google Nest Hub Max з підтримкою Google Assistant є нашими фаворитами для вітальні, тоді як менші варіанти більше підходять для кухні та спальні.

Один із найпростіших і найдоступніших способів оновити вітальню – це замінити стандартні лампи розжарювання на **розумні лампочки,** які підтримують програмне та голосове керування. Після оновлення вам більше ніколи не доведеться вставати з дивана, щоб знову натиснути вимикач. Окрім більшої зручності, ніж традиційні лампи, розумні світильники споживають менше енергії та служать довше, тому вони можуть заощадити ваші гроші в довгостроковій перспективі.

Розумна лампочка AiDot Linkind Wi-Fi Matter оснащена вбудованим Wi-Fi і працює з Matter, тому ви можете легко підключити її до обраної платформи розумного дому, будь то Alexa, Google, HomeKit або SmartThings. Він відображає 16 мільйонів кольорів і діапазон білих тонів, а також пропонує власні та музичні ефекти, щоб допомогти вам створити настрій для вечора кіно.

Ще один розумний гаджет для вітальні, який може додати серйозних

КОПІЮВАННЯ

зручностей у ваше життя та потенційно заощадити гроші на електроенергії, – це **підключений термостат**. Наші улюблені моделі дозволяють контролювати та контролювати систему опалення та охолодження вашого дому за допомогою телефону чи голосу, встановлювати температурні графіки та пропонують функції енергозбереження, які можуть допомогти знизити ваш рахунок. Наш найкращий вибір, Ecobee Smart Thermostat Premium, оснащений дистанційним датчиком, який можна розмістити в іншій кімнаті, щоб забезпечити постійну температуру в усьому будинку. Як більш бюджетний варіант ми настійно рекомендуємо Amazon Smart Thermostat.

За допомогою **розумної розетки** ви можете перетворити будь-що, що підключається до стіни, – лампу, обігрівач, зволожувач повітря чи навіть новорічну ялинку – на підключений пристрій, яким можна керувати за допомогою телефону. TP -Link Kasa Smart Wi-Fi Plug Slim With Energy Monitoring, наш поточний вибір редакції в цій категорії, відстежує споживання електроенергії будь-яким підключеним вами пристроєм, працює з голосовими командами та підтримує безліч сторонніх платформ, і все це без збитків. TP -Link Kasa Smart Wi-Fi Power Strip HS300 – це розумний фільтр від перенапруг із шістьма розетками, якими можна незалежно керувати за допомогою телефону або голосу. Він також підтримує IFTTT, тож ви можете запрограмувати його на автоматичне вмикання або вимикання певних розеток, коли відбувається інша подія, наприклад тригер руху камери.

Оскільки у вашій вітальні так багато розумних гаджетів, гарною ідеєю буде інвестувати в **домашню камеру безпеки,** підключену до Wi-Fi. Одна з наших улюблених відеокамер Eufy S220 Indoor Cam забезпечує яскраве відео високої роздільної здатності та оснащена функціями, типовими для дорожчих моделей, включаючи локальне та хмарне зберігання, автоматичне відстеження руху та підтримку всіх основних платформ голосової допомоги.

Нарешті, якщо у вас є кіт або собака, ваша розумна вітальня була б неповною без **камери для тварин**, яка дозволяє стежити за своїми пухнастими

КОПІЯ 2025

малюками та взаємодіяти з ними 24/7, де б ви не були. Камера для собак Furbo 360 така ж функціональна, як і весела, пропонуючи сповіщення про гавкіт і функцію дистанційного підкидання ласощів, яка сподобається вам і вашій собакі. Щоб порадувати своїх котячих сусідів по кімнаті, виберіть камеру для котів Furbo 360, яка в основному така ж, як і версія для собак, але пристосовує звукові ефекти та сповіщення до котячих, додаючи вгору іграшку з пір'ям.

Кухня: необхідні підключені інструменти для приготування їжі та прибирання

Незалежно від того, чи ви шеф-кухар, чи досвідчений користувач Uber Eats, найкращі розумні кухонні гаджети можуть принести серйозну зручність у серце вашого дому.

Не можете вирішити, що на вечерю? Потрібно швидко перетворити рідкі унції на чашки? Шукаєте заміну кінзи? Схоже, вам потрібен **розумний дисплей** на стільниці. За допомогою Alexa або Google Assistant на вашій кухні ви також можете швидко встановлювати таймери, додавати продукти до списку покупок і відтворювати улюблену кулінарну музику, і все це лише за допомогою голосу. Echo Show 8 третього покоління від Amazon – наш улюблений компаньйон на кухні; він якраз підходить для прилавків, розважить вас під час роботи з чудовими динаміками, а також може здійснювати пошук і проводити вас через численні рецепти для приготування їжі.

Коли ви визначитеся з рецептом, найкраща **інтелектуальна кухонна техніка** допоможе вам на кожному кроці процесу приготування. Розумна піч Tovala дозволяє випікати, смажити, смажити та готувати їжу на пару за допомогою програмного керування та QR-кодів, що робить її однією з найрозумніших настільних печей, які ви можете купити. Додаткові набори страв Tovala дорогі, але смачні, а багатоетапні програми приготування допоможуть вам приготувати перші страви рівня шеф-кухаря.

Зрештою, є розумна версія практично будь-якого іншого кухонного приладу, який тільки можна придумати, від повнорозмірних духовок і

КПЗ-2025

холодильників до менших гаджетів, таких як фритюрниці та Instant Pot Pro Plus. Ви навіть можете задовольнити свою тягу до льоду з самородків за командою за допомогою льодогенератора Ecozy Smart Nagget Ice Maker із підтримкою Wi-Fi.

І, звичайно, жодна кухня не обійдеться без кавоварки. У ідеальному розумному домі підключено навіть вашу машину для варіння зерен. **Розумні кавоварки** дозволяють керувати додатками та голосом, тому єдине, що стоїть між вами та поштовхом Java, – це кілька натискань на вашому телефоні або фраза: «Алекса, увімкни мою кавоварку». Один із таких варіантів, розумна кавоварка Hamilton Beach, дозволяє використовувати голосові команди, щоб увімкнути та вимкнути її та встановити силу заварювання, і вона підтримує процедури Alexa, щоб ви могли запрограмувати її на автоматичне приготування кави щодня в певний час. Якщо ви можете витратити більше, розумна кавоварка xBloom Studio автоматизує помел і заварювання з точністю на рівні бариста, результатом чого стане ретельно виготовлена чашка кави.

Коли прийде час прибирати після обіду, хороший **робот-пилосос** подбає про підлогу замість вас. Одна з наших улюблених моделей, Shark IQ Robot Self-Empty XL R101AE, працює швидко та ефективно, рідко застряє та має великий смітник у базовій станції, який потрібно спорожняти лише раз на місяць. Як більш бюджетний варіант, доступний TP-Link Tapo RV30 Max Plus вразив нас своєю ефективною навігацією та самовипорожненням смітника, обидві функції є рідкісними для його ціни менше 300 доларів.

Ви також можете розглянути преміальний **робот-пилосос із автоматичним спорожненням,** як-от Ecovacs Deebot X2 Omni, кращий універсальний робот для прибирання вашої підлоги. У ньому є сміттєзбірник, який самостійно спорожняється, швабра, яка сама очищає та висихає, а також вбудований голосовий помічник. Для будинків з домашніми тваринами ми настійно рекомендуємо iRobot Roomba j9+, який поставляється з гарантією POOP (офіційна обіцянка власника домашніх тварин): якщо робот не зможе уникнути твердих відходів домашніх тварин протягом року після покупки, компанія

КПЗ-2025

безкоштовно подарує вам новий.

Щоб відтерти смуги, вам потрібна хороша **швабра-робот**. Bissell SpinWave Wet and Dry оснащено двома обертовими накладками з мікрофібри та підтримує кілька різних розчинів для чищення, щоб ваші підлоги виглядали як нові. iRobot Braava Jet m6 – ще один чудовий варіант, який підмітає та миє вашу підлогу до блиску.

Нарешті, якщо ви знаєте про мікроби, розумне відро для сміття, яке може відкривати кришку, може вас зацікавити. Прямокутний сенсорний контейнер із подвійним відділенням Simplehuman, який ми не перевіряли, оснащений боковими баками для сміття та переробки та реагує на голосові команди, такі як «відкрити банку», «залишатися відкритим» і «закрити банку».

Спальня: техніка для кращого нічного сну

Якщо ваше ліжко занадто незручне, у вашій кімнаті занадто жарко, або ви відчуваєте труднощі з диханням через алергію чи астму, найкраща техніка для спальні може запропонувати деяке полегшення.

Для максимального нічного комфорту ліжко 360 i8 Smart Bed від Sleep Number – це значні інвестиції, але воно дає змогу регулювати міцність кожної сторони незалежно, тож якщо ви любите спати на хмарі, а ваш партнер віддає перевагу цеглині, у вас обох буде все по-своєму. Він також відстежує ваші тенденції сну в супутньому додатку та пропонує персоналізовану статистику, яка допоможе вам краще дрімати.

Температура у вашій кімнаті також може впливати на якість сну. У надто жаркій або надто холодній кімнаті може бути важко заснути й заснути протягом ночі. За даними CDC, ідеальною температурою для сну для більшості людей є приблизно від 65 до 68 градусів за Фаренгейтом. Розумний матрац-топпер Eight Sleep Pod Pro Cover дозволяє регулювати температуру наявного ліжка від гарячої до прохолодної за допомогою телефону та експертно відстежує стан вашого сну.

Якщо у вас немає центрального кондиціонування, ви можете розглянути **розумний кондиціонер**. Зазвичай вони дозволяють керувати додатками й

КОПЗ-2025

голосом, а також підтримують планування, що може допомогти скоротити витрати на електроенергію. Розумний кондиціонер GE Profile Clearview охолоджує швидко й тихо завдяки інноваційній конструкції розділеного корпусу, яка розміщує основну частину пристрою під лінією вікна. У результаті найгучніші шматки залишаються на вулиці, а ви все ще можете бачити з вікна.

Розумний **очищувач повітря** може допомогти вам контролювати якість повітря в приміщенні та дихати трохи легше, особливо якщо ви страждаєте на алергію та астму. Dyson Big+Quiet Formaldehyde BP03, наша улюблена модель для всіх, є одночасно потужним триступеневим очищувачем повітря HEPA, який може позбавити ваш дім від шкідливих забруднювачів, і обертовим вентилятором із підключенням Wi-Fi для телефонного та голосового керування.

Для розслаблюючого **розумного освітлення** настільна лампа Philips Hue Iris підтримує до 16 мільйонів кольорів і процедур, щоб допомогти вам прокинутися або заснути. Ви можете запланувати його вимкнення після того, як ви заснете, або просто повідомити йому, що ви готові лягати спати, за допомогою голосових команд Amazon Alexa або Google Assistant. Його легко налаштувати, а його максимальна потужність 570 люменів дає змогу ставати гарним і яскравим, коли ви цього хочете.

Розумний **динамік** може стати чудовим компаньйоном біля ліжка. Наші улюблені моделі для спальні – це Amazon Echo Spot і Google Nest Mini. Вони не будуть переповнювати вашу тумбочку та дозволять вам встановлювати будильники, відтворювати музику, перевіряти погоду та керувати іншими сумісними пристроями розумного дому – і все це за допомогою голосу.

Подібним чином **розумний дисплей** пропонує всі переваги розумної колонки та організовує елементи керування розумним домом на одному екрані, зручно розташованому на тумбочці біля ліжка. Nest Hub другого покоління достатньо малий для тумбочки та оснащений датчиками, які дозволяють йому відстежувати ваш сон, частоту дихання вночі, кашель і хропіння, а також будь-які зміни освітленості та температури у вашій кімнаті, і все це без камери або

КПЗ – 2025

необхідності носити окремий пристрій у ліжко. Потім він пропонує корисну інформацію, яка допоможе покращити ваше закрите око. Якщо ви віддаєте перевагу Alexa, наш улюблений розумний дисплей для спальні – це Echo Show 5 третього покоління. Він не відстежуватиме ваш сон, але має камеру (яку за бажанням можна вимкнути) для відеодзвінків і віддаленого спостереження за домом.

Нарешті, якщо у вас є маленькі діти, **розумний радіоняня** дозволить вам дивитися, розважати та заспокоювати дитину в її ліжечку. Повна система моніторингу Nanit Pro вимагає глибоких кишень, але це найкращий інтелектуальний радіоняня, який ми тестували, особливо для тих, хто вперше та/або нервових батьків. Він записує час сну та пробудження вашої дитини з точністю до хвилини, що може допомогти вам звузити час сну для вашого маленького херувима. Він також може попередити вас, якщо ваша дитина перестає дихати, щоб запобігти ймовірності синдрому раптової дитячої смерті.

Ванна кімната: гаджети для догляду за ранком без стресу

Вода та технології зазвичай не змішуються, але це не зупинило зростання кількості підключених гаджетів для ванних кімнат, від високотехнологічних унітазів, які можуть очищатися самостійно, до розумних ваг, які можуть вимірювати набагато більше, ніж просто вашу вагу.

Якщо ви будуєте новий будинок або ремонтуєте старий, подумайте про розумний туалет для вашої ванної кімнати. Ми ще не розглядали жодного, але Kohler Veil пропонує систему відкривання та закривання без використання рук, яка має усунути суперечки щодо того, залишати сидіння унітазу піднятим чи опущеним. Він також має нічник, сидіння з підігрівом, систему сушіння теплим повітрям, автоматичний дезодорант, функцію самоочищення, яка використовує ультрафіолетове світло для дезінфекції чаші, і аварійний режим, який дозволяє змивати воду до 100 разів під час відключення електроенергії. Якщо ви хочете покращити наявний унітаз, Kohler PureWash E930 – це розумне біде, яке дозволяє попросити Alexa промити ваш зад.

КПЗ-2025

У ідеальній розумній ванній кімнаті ви навіть можете запустити душ своїм голосом. U by Moen – це настінний розумний контролер для душу, який працює з Amazon Alexa, Apple HomeKit і Google Assistant, щоб дозволити вам увімкнути воду та встановити бажану температуру за допомогою голосових команд.

Покращіть своє марнославство за допомогою Simplehuman Sensor Mirror Hi-Fi, який має різноманітні настроювані параметри освітлення для бездоганного нанесення макіяжу та вбудований динамік, щоб ви могли слухати улюблену музику або подкасти, готуючись. Він підсвічується автоматично, коли виявляє ваше обличчя, і підтримує голосове керування Alexa, функції, які стають у нагоді, коли в одній руці ви тримаєте пудру, а в іншій – пензлик.

Розумні **ваги** корисні, якщо ви хочете скинути кілька кілограмів або просто хочете уважніше стежити за своїм здоров'ям. Ваги Etekcity HR Smart Fitness Scale відстежують 14 показників, включаючи ваш ІМТ, відсоток жиру в організмі, відсоток води в організмі, частоту серцевих скорочень, м'язову масу, рівень білка та вагу до 400 фунтів. Він також має вбудований Wi-Fi та акумуляторну батарею, і все це за розумну ціну, що заслужило нашу головну рекомендацію.

Наша улюблена технологія підключеної ванної може навіть допомогти вам отримати A+ під час наступного візиту до стоматолога. Електрична зубна щітка Laifen Wave має достатню потужність, щоб добре почистити зуби, і вона працює з додатком, який дозволяє налаштувати силу вібрації, діапазон коливань і швидкість коливань.

Домашній тренажерний зал: інтерактивні тренажери для досягнення ваших цілей

Незалежно від того, чи пощастить вам мати спеціальну кімнату для тренувань чи просто порожній куточок із достатнім місцем для килимка для йоги, ринок **обладнання для тренажерного залу для розумного дому** не запропонує брак варіантів для роботи над вашою фізичною формою.

Інтелектуальні фітнес-пристрої, такі як Peloton Bike+, забезпечують рівень

КПЗ_2025

інтерактивності, що набагато перевищує їх більш доступні аналоги без зв'язку. Bike+ і багато інших пропонують велику бібліотеку занять із досвідченими інструкторами та/або персоналізовані тренування та програми на основі ваших цілей і поточного рівня фізичної підготовки. Це як особистий тренер у комфорті, безпеці та приватності власного дому.

Їзда на велосипеді в приміщенні може допомогти вам спалити калорії, підвищити витривалість серцево-судинної системи та підняти настрій. Ви можете припустити, що Peloton є **найкращим розумним стаціонарним велосипедом,** оскільки він найпопулярніший, але насправді найкращий варіант – це той, який мотивуватиме вас повертатися на нього день за днем. Peloton мотивує вас за допомогою показників і конкуренції, а SoulCycle At-Home Bike, ще один переможець вибору редакції в цій категорії, пропонує більш медитативний досвід, орієнтований на танці. Як більш доступний варіант ми також настійно рекомендуємо BODi Bike (раніше MYX II), який пропонує широкий спектр тренувань і наголошує на персоналізованому тренуванні зон серцевого ритму.

Для тренування всього тіла з низьким навантаженням, яке мінімально навантажує коліна та інші суглоби, розгляньте **розумний гребний тренажер**. NordicTrack RW900 є відмінним вибором для початківців, середніх і досвідчених веслувальників. Він має 22-дюймовий сенсорний екран, на якому ви можете переглядати та транслювати тренування iFit на відкритому повітрі, зняті в мальовничих місцях по всьому світу, від озера Інніскарра в Ірландії до річки Кафуе в Замбії. Під час тренувань на відкритому повітрі тренажер автоматично регулює ваш опір відповідно до умов вітру та води, щоб ви відчували те, що відчуває тренер. Тренери часто вказують на визначні пам'ятки дорогою, тож ви отримаєте тренування *та* екскурсію. Hydrow дорожчий, але це один із найрозкішніших розумних гребних тренажерів, які ви можете придбати, поєднуючи в собі найсучасніше обладнання та пряму трансляцію тренувань.

У той час як гребні тренажери можуть лякати новачків, а велосипедні сидіння можуть бути буквальним болем у сідниці, бігові доріжки є більш

КПІ – 2025

зручними, доступними та зручними, оскільки ходьба – це діяльність, яку більшість із нас все одно виконує щодня. Peloton Tread займає приблизно стільки ж місця, скільки диван, і має чудовий 23,8-дюймовий сенсорний екран 1080p, за допомогою якого ви можете отримати доступ до величезної та різноманітної бібліотеки Peloton добре створених, розважальних занять. Для більш доступного варіанту ми рекомендуємо Echelon Stride, який підключається до вашого смартфона або планшета, щоб ви могли транслювати заняття з бігу та ходьби в прямому ефірі та на вимогу з конкурентоспроможними таблицями лідерів і складатися, коли не використовуєте.

Щоб наростити м’язи, подумайте про **розумний силовий тренажер**. Нашим улюбленим є Tempo Studio, який має дизайн, схожий на мольберт, із 42-дюймовим сенсорним екраном, на якому можна транслювати різноманітні тренування під керівництвом тренера, у тому числі заняття в прямому ефірі. Під екраном знаходиться шафа, в якій акуратно зберігаються чотири набори гир, тому вони не видно, коли вони не використовуються. На спину можна повісити гантелі і штангу, що входять в комплект. Тренажер відстежує, яку вагу ви повинні підняти, підраховує повторення та пропонує відгуки про вашу форму, щоб допомогти вам безпечно тренуватися. Tonal, ще один чудовий варіант у цій категорії, використовує комбінацію електрики та магнітів для оцифровки гир, пропонуючи опір до 200 фунтів у гладкій настінній машині. Водночас камера Peloton Guide пропонує допомогу в силових тренуваннях за набагато меншу суму. Для цього потрібен лише телевізор і трохи вільного місця.

Не вистачає місця? Спробуйте Forme Studio. Коли не використовується, виглядає як звичайне дзеркало. Коли прийде час тренуватися, ви можете використовувати повний сенсорний екран і персоналізовані тижневі плани тренувань від фітнес-консьєржа, щоб допомогти вам досягти своїх цілей.

Гараж: спроектуйте автоматизований та організований багатоцільовий простір

Незалежно від того, чи служить ваш гараж місцем для зберігання,

КОПЗ – 2025

майстернею, зоною для вечірок або місцем, де ви фактично зберігаєте свій автомобіль, є багато доступних технологій, щоб зробити його розумнішим.

Якщо ваш старий механізм відкривання дверей гаража насилу піднімає та опускає двері, і при цьому шумить, можливо, настав час для оновлення. Інтелектуальні механізми відкривання гаражних воріт із ремінним приводом зазвичай працюють безшумно, порівняно з моделями з ланцюговим приводом минулого. Що ще важливіше, вони дозволяють переглядати стан дверей (зачинені чи відкриті, і як довго) і контролювати їх зі свого телефону, тож вам не доведеться плестися на вулицю в піжамах, коли вам уже зручно в ліжку. Genie QuietLift Connect надзвичайно тихий і підключається до вашої домашньої мережі Wi-Fi, дозволяючи вам керувати ним за допомогою телефону або планшета. Він підтримує голосові команди Amazon Alexa та Google Assistant і відносно простий в установці завдяки рейковій системі з п’яти частин, які з’єднуються разом.

Пристрої для відкривання дверей гаража, які працюють із додатками Aladdin Connect, myQ або OHD Anywhere, сумісні зі службою доставки в гаражі Amazon, Key, яка може дати вам спокій, якщо ви хвилюєтеся, що пірати викрадуть пакунки з ваших вхідних дверей. Щоб налаштувати його, зв’яжіть сумісні двері гаража з Amazon Key у програмі Amazon і ввімкніть доставку в гараж. Послуга ексклюзивна для учасників Amazon Prime; коли ви це налаштуєте, просто виберіть «Доставка в гараж» під час оформлення замовлення. Раніше це була безкоштовна пільга Prime, але тепер Amazon стягує 1,99 доларів США за доставку в гараж за межі робочих місць доставки Amazon Day.

У вас немає грошей на нову розумну систему відкривання дверей гаража? Нічого страшного: доступний Chamberlain Smart Garage Hub дає змогу додати підключення до існуючої системи. Він сумісний з усіма основними механізмами відкривання гаражних воріт, виробленими після 1993 року.

На додаток до високотехнологічного автомобіля та електровелосипеда, гараж, ймовірно, є місцем, де ви зберігаєте багато своїх **пристроїв для розумного дому на вулиці**. Один із наших фаворитів, інтелектуальний

КОПЗ-2025

спринклерний контролер Rachio 3, допоможе вам підтримувати зелений і пишний вигляд вашого улюбленого газону, заощаджуючи гроші на воді. Він встановлюється за лічені хвилини та дає вам змогу керувати вашою системою спринклерної системи з будь-якого місця за допомогою телефону. Він також може автоматично перепланувати цикли поливу, коли очікується значна кількість опадів. Це видатна функція, яка показала хороші результати в наших тестах.

Наша екскурсія найкращим розумним будинком була б неповною без відвідування заднього двору. Найкращі **пристрої для розумного дому на відкритому повітрі** можуть допомогти вам зберегти зелений двір, блакитний басейн і безпеку вашого майна. Вони також дозволяють легко створити теплу та привабливу атмосферу для вечірок у дворі.

Щоб створити настрій для вашого наступного барбекю або вечірки біля басейну, вам потрібен хороший **зовнішній динамік**, який витримує бризки та дощ. Вибирайте Sonos Move 2, якщо вам потрібен потужний і надійний динамік. Він підтримує стереозвук, має тривалий час роботи від батареї та має надійний рейтинг захисту від атмосферних впливів IP56. Він також пропонує голосове керування в режимі hands-free за допомогою Alexa або основного асистента Sonos.

Якщо вам не потрібне голосове керування на внутрішньому дворику, виберіть Sony ULT Field 7, щоб отримати наше улюблене поєднання розміру та якості звуку, або спробуйте наш улюблений бюджетний варіант Anker Soundcore Motion 300.

Плануєте кіновечір у своєму дворі? Зовнішній **проектор** може перетворити зовнішню стіну на кінотеатр. Anker Nebula Mars 3 є нашим найкращим вибором за його якість зображення та довговічність, а AAXA P6 Ultimate є нашим улюбленим недорогим зовнішнім проектором.

Завдяки **розумним світильникам**, які зручно використовувати на вулиці, вечірці не потрібно зупинятися, коли заходить сонце. Веселі та святкові ліхтарі

КОПІЯ 2025

Govee Lynx Dream Bluetooth & Wi-Fi Outdoor String Lights розроблені для внутрішнього дворика з атмосферостійкою конструкцією та простими в установці. Ви можете керувати цими різнокольоровими вогнями за допомогою голосу за допомогою Alexa або Google Assistant або через їх добре розроблений супутній додаток. Govee пропонує кілька попередньо встановлених сцен, щоб створити настрій, або ви можете використовувати режим Music Sync для вечірок.

Якщо у вас є ланцюгові світильники або декоративна водяна споруда, яку ви хочете покращити, вам може допомогти **розумна розетка,** призначена для використання на вулиці. Захищена від атмосферних впливів зовнішня розетка TP-Link Kasa EP40A працює з усіма основними платформами розумного будинку, включаючи Alexa, Google Assistant, HomeKit, IFTTT і SmartThings. Він має міцну конструкцію та два розетки, якими можна керувати незалежно.

Якщо ви справді хочете справити враження на своїх друзів на наступному вечірці у дворі, перегляньте наш список найкращих **гаджетів для гриля та барбекю**. Traeger Ironwood 650 може допомогти вам готувати барбекю, смажити та коптити з деревними гранулами, як професійний пітмейстер. Kamado Joe Konnected Joe Digital Charcoal Grill and Smoker є найкращим для гриля на деревному вугіллі, тоді як Weber Genesis EPX-335 Smart Gas Grill є найкращим газовим варіантом.

Для приготування на грилі за допомогою штучного інтелекту зверніть увагу на Brisk It Origin 940. Це гриль на деревних гранулах, який дозволяє смажити, коптити, пекти та готувати барбекю без присмаку рідини для запальничок. Окрім основ, механізм штучного інтелекту під назвою Vera пропонує незліченну кількість рецептів, які ви можете вибрати та відправити на гриль. Кожен рецепт, який ми спробували, привів до приголомшливих результатів нашого тестування.

Якщо у вас немає грошей на смарт-гриль, термометр Weber iGrill Mini дозволяє контролювати температуру їжі зі смартфона. Це дозволить вам розслабитися та провести час з друзями та сім’єю замість того, щоб весь час

КПІ – 2025

зависати над грилем.

Нарешті, щоб отримати найкращий газон у кварталі, розгляньте роботу - **газонокосарку,** таку як Husqvarna Automower 435X AWD, яка легко справляється з важкою місцевістю та працює з голосовими командами Alexa та Google. Щоб отримати більш доступний варіант, ознайомтеся з газонокосаркою-роботом Gardena Sileno City, яка пропонує легке планування та тиху роботу. І якщо набридливі комарі заважають вам насолоджуватися свіжоскошеним газоном, подумайте про придбання розумної системи відлякування комарів Thermacell LIV.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Оскільки потрібно розробити просту та легку у користуванні програму, яка б виконувалась під операційною системою Windows, то для її реалізації я обрав Builder C++. Існує велике число бібліотек написаних під Builder C++ , тому це одна з важливих причин вибору мови програмування. Середовище Builder C++ досить просте в користуванні, його вихідний код значно менше по об’єму в порівнянні з Delphi чи деякими іншими програмами такого типу. Досить легко організувати взаємодію між модулями програм, об’єктно-орієнтований підхід дає можливість значно скоротити код програми, а отже і час його виконання.

На заміну старого розробленого набору елементів управління у Builder C++ інтегрована бібліотека візуальних компонентів VCL, представлених на палітрі компонентів. Після переносу на форму методом перетягування (drag-and-drop) компоненти відразу становляться діючими об’єктами вашої програми. Окрім типізованих інтерфейсних елементів Windows (кнопки, смуги прокручування, редагуємі текстові області, прості та комбіновані списки, та інше) у бібліотеку включені елементи підтримки діалогових вікон, обслуговування баз

КПЗ - 2025

даних та багато іншого. Можливо не тільки модифікувати поведінку існуючих компонентів, але і будувати нові.

Builder C++ підтримує останні розширення стандарту мови C++ та забезпечує швидку компіляцію та складання 32-розрядних програм для Windows. Результуючі програми оптимізовані з точки зору швидкості виконання програм та затрат пам’яті. Зручний відладгоджувальник (з асемблерним вікном, можливістю крокового виконання, завдання точок зупинки, трасування та інше) повністю інтегрований у систему проектування. Дизайнер форм, редактор коду, інспектор об’єктів та інші інструменти зостаються доступними під час виконання програми, саме через це вносити зміни до коду можна прямо у процесі відлагодження. Дизайнер форм, Інспектор об'єктів і інші засоби залишаються доступними під час роботи програми, тому вносити зміни можна в процесі відлагодження. Builder C++ поставляється в трьох варіантах: Standard (стандартний), Professional (для професіоналів розробників, орієнтованих на мережеву архітектуру) і Client/Server Suite (для розробки систем в архітектурі клієнт/сервер). Останні два варіанти доповнюють стандартний початковими текстами візуальних компонентів, різномасштабним словником даних, новими функціями мови запитів SQL для бази даних, пакетом підтримки систем Internet, службою моніторингу програм, а також рядом інших засобів.

Builder C++ підтримує зв'язок з різними базами даних 3-х видів: dBASE і Paradox: Sybase, Oracle, InterBase і Informix; Excel, Access, FoxPro і Btrieve. Механізм BDE (Borland Database Engine) додає обслуговуванню зв'язків з базами даних дивовижну простоту і прозорість. Провідник Database Explorer дозволяє зображати зв'язки і об'єкти баз даних графічно. Використовуючи компоненти баз даних, я побудував електронний записник згідно таблиці dBASE за півгодини роботи на комп'ютері. Спадкоємство готових форм і їх "підгонка" під специфічні вимоги помітно скорочують тимчасові витрати на вирішення подібних завдань.

Довідкова служба Builder C++ надавала мені допомогу в цій і багатьох інших подібних ситуаціях. Є повний опис кожного управляємого компонента,

КОПЗ-2025

включаючи списки властивостей і методів, а також численні приклади. Виклад матеріалу в книзі був значно покращуваний і систематизований завдяки відомостям, почерпнутим мною з довідкової служби.

Завдяки засобам управління проектами, двосторонній інтеграції додатку і синхронізації між засобами візуального і текстового редагування, а також вбудованому відладнику (з асемблерним вікном прокрутки, покрокового виконання, точок останову, трасуванням і тому подібне) – Builder C++ корпорації Borland надає собою вражаюче середовище розробки, яка, мабуть, витримає конкурентну боротьбу з такими модними продуктами як Developer Studio фірми Microsoft.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи IoT інтелектуального будинку з використанням мережі LPWAN.

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

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації

КПЗ_2025

в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

КбПЗ_2025

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Зі швидким розвитком технологій розширюється практичне застосування автоматизації розумного дому, що робить наше життя комфортнішим і зв’язаним. У цій статті ми розглянемо кілька реальних прикладів автоматизації розумного дому та розглянемо, як вони змінюють повсякденне життя.

Автоматизація розумного будинку:

1. Керування освітленням за допомогою голосу: уявіть, що ви входите в будинок і просто говорите: «Алекса, ввімкни світло». і це працює!

2. Автоматизація термостата може допомогти вам автоматично регулювати температуру та таким чином оптимізувати споживання енергії. Ви можете керувати своїм термостатом дистанційно зі свого смартфона, забезпечуючи комфорт у вашому домі, коли ви приїдете, і заощаджуючи витрати на електроенергію.

3. Безпека та відеоспостереження: розумні системи домашньої безпеки пропонують низку функцій, від відеодзвінків у двері, які дозволяють віддалено бачити та спілкуватися з відвідувачами, до камер, що активуються рухом, які надсилають сповіщення на ваш телефон. Ці системи підвищують безпеку будинку, забезпечуючи душевний спокій, коли вас немає.

4. Контроль розваг: за допомогою автоматизації розумного дому ви можете централізувати керування розважальною системою.

5. Виявлення витоку води: Розумні детектори витоку води можуть виявляти витік або надлишок вологи та негайно надсилати сповіщення на ваш смартфон.

6. Здоров’я та самопочуття: такі пристрої, як розумні ваги, монітори АТ і засоби відстеження сну, можуть надавати дані, які допоможуть вам вести

КПЗ_2025

здоровий спосіб життя.

Підсумовуючи, розумна автоматизація будинку – це не просто розкіш, а практичний і доступний спосіб покращити якість повсякденного життя.

Освітлення

Системи розумного освітлення забезпечують дистанційне централізоване керування всіма освітленнями в будинку, дозволяючи користувачам налаштовувати автоматизацію та вмикати чи вимикати світло за допомогою програми на смартфоні. Інтеграція систем освітлення з розумним помічником, таким як Amazon Alexa або Google Assistant, дозволяє користувачам взаємодіяти з лампочками за допомогою голосу. Наприклад, систему можна налаштувати на автоматичне вимикання світла, коли нікого немає вдома. Навпаки, датчики IoT можуть увімкнути світло, якщо відчують, що в кімнаті занадто темно. Крім того, системи освітлення можуть слідувати певним шаблонам, наприклад, приглушувати світло в спальні після 23:00 або в їдальні, коли ввімкнено телевізор.

HVAC

Розумні датчики термостата вимірюють температуру та вологість у домі та передають дані системам опалення, вентиляції та кондиціонування повітря (HVAC). Системи ОВК можуть підтримувати температуру на комфортному рівні для всіх мешканців. Користувачі також можуть запрограмувати системи HVAC для підвищення температури за годину до повернення людей додому. Крім того, HVAC може автоматично регулювати температуру залежно від активності мешканців: якщо людина займається йогою чи фізичними вправами, температура автоматично знизиться.

Камери спостереження

Камери відеоспостереження забезпечують постійне спостереження за приватними будинками та квартирами в реальному часі. Вони цілодобово записують усе всередині та поза домом і передають дані в хмару. За допомогою камер користувачі можуть перевіряти своїх домашніх тварин або людей похилого

КОПЗ – 2025

віку, які залишаються одні протягом дня. У разі вторгнення в будинок мешканці матимуть повний запис, який може бути корисним поліції.

Системи безпеки

Розумні датчики безпеки можуть відстежувати витоки газу чи водопроводу та сповіщати користувачів за допомогою мобільних додатків. Коли користувач отримує сповіщення, він може дистанційно вимкнути воду в магістралі або викликати аварійну службу, щоб запобігти небезпечним наслідкам. Крім того, вікна можуть бути обладнані іншими типами розумних датчиків для відстеження всіх дій відкриття та закриття в будинку.

Розумні замки

Вхідні двері, оснащені розумними замками, дозволяють людям входити в приміщення не тільки зі стандартними ключами, але й з кодами, біометричними даними або функцією розпізнавання обличчя. За допомогою підключеного мобільного додатку користувачі можуть дистанційно впускати своїх гостей у будинок і завжди бути в курсі всіх переміщень вхід і вихід. Крім того, жителі будинку можуть підключити розумні замки до систем освітлення та опалення, щоб вмикати необхідні прилади, щойно відчиняються двері.

3.2 Розробка структурної схеми

Для побудови мереж LPWAN уже розроблена безліч технологій, у тому числі в Україні, однак у глобальному масштабі трьома основними вважаються LoRa, Sigfox і NB-IoT.

При розгортанні мережі LoRaWAN можуть використовуватися як вишки стільникового зв'язка, так і інші місця установки: наприклад, промислові шлюзи можна розміщати на виробничих площадках, а невеликі пікошлюзи – прямо на будинках. Вартість шлюзу для розміщення на вишці оцінюється в 1000 доларів, промислового шлюзу – приблизно в 500, шлюз для пікосоти може коштувати

КБПЗ-2025

всього близько 100 доларів. Витрати на модернізацію базової станції 4G LTE для підтримки NB-IoT, по оцінках експертів, можуть скласти більше 10 тис. доларів.

Специфікація NB-IoT опублікована в червні 2016 року, однак випуск перших мікросхем для кінцевих пристроїв з підтримкою цієї технології запланований тільки на 2027 рік. Потім буде потрібно час на тестування, організацію масового виробництва продуктів NB-IoT, запуск мереж у комерційну експлуатацію. Інакше кажучи, пройде ще пари-трійка років доти, поки NB-IoT стане реальністю. Рішення LoRa уже доступні. Вони випробувані, а відповідні мережі перебувають у комерційній експлуатації в ряді країн, у тому числі в національному масштабі.

Як відзначають фахівці, новий радіоінтерфейс EC-GSM-IoT дозволить багаторазово збільшити кількість працюючих у мережі пристроїв M2M, а за рахунок росту чутливості радіомодулів удасться в сім разів розширити радіус дії базової станції й знизити енергоспоживання пристроїв. Крім того, затверджується, що впровадження EC-GSM-IoT не зажадає масштабної заміни встаткування зв'язку – у більшості випадків можна обійтися відновленням ПЗ на вузлах радіомережі.

LPWAN (англ. Low-power Wide-area Network – «енергоефективна мережа далекого радіуса дії») – бездротова технологія передачі невеликих по обсязі даних на далекі відстані, розроблена для розподілених мереж телеметрії, міжмашиної взаємодії й інтернету речей. LPWAN є однієї з бездротових технологій, що забезпечують середовище збору даних з різного встаткування: датчиків, лічильників і сенсорів [1].

Застосування:

– Енергетика: системи бездротового автоматизованого контролю й обліку електроенергії [5].

– Житлово-комунальне господарство: віддалений збір показань приладів обліку: води, тепла, газу, електрики [6]

КОПЗ_2025

– Сільське господарство: контроль вологості й температури ґрунтів, моніторинг освітленості й рівня сонячної радіації, моніторинг стану складів і овочесховищ.

– Безпека й охоронні системи: резервування каналів зв'язку, контроль проникнення, контроль затоплення приміщень, контроль пожежної безпеки.

– Системи міського менеджменту й «розумне місто»: керування освітленням [7], моніторинг автомобільного трафіку, контроль зайнятості парковочних місць, контроль температури, шуму, вологості, освітленості, забруднення повітря й т.п.

– Моніторинг навколишнього середовища: контроль якості повітря й води, контроль рівня води в ріках і озерах, моніторинг пожежної небезпеки в лісах.

– Будівництво: контроль устаткування, моніторинг параметрів конструкцій і будинків.

– Медицина: пристрої, що носяться.

– Автотранспорт: контроль швидкісного режиму й стилю водіння.

– Виробництво й системи постачання: контроль параметрів устаткування, моніторинг стану вантажів.

– Додатка інтернету речей [1].

На рисунку 3.1 зображена узагальнена структурна схема системи.

Як було відзначено вище, сучасний інтелектуальний будинок – це дуже складне інженерне рішення, що складається з наступного набору систем:

– Система безпеки.

– Система комфорту.

– Інформаційна система.

– Система диспетчеризації.

КПТВ_2025

Рисунок 3.1 – Структурна схема системи

Розглянемо які підсистеми входять у перераховані вище системи.

Система безпеки:

– Система цифрового відеоспостереження з можливістю одночасного спостереження, перегляду, архівування.

– Режим віддаленого перегляду й управління через Інтернет.

– Бездротова пожежна й охоронна сигналізація з можливістю обміну інформацією через GSM модуль.

– Система контролю доступу в приміщення (у тому числі віддалене управління гаражними воротами).

Система комфорту:

– Внутрішня телефонна система, голосний зв'язок усередині будинку.

КОПЗ – 2025

– Система супутникового, ефірного телебачення з можливістю перегляду в будь-якій кімнаті.

– Система "Домашній кінотеатр".

– Система "Мультирум" аудіо– й відео– (система "звук навколо").

– Цифровий розважальний комплекс, оцифровка відео, печать фотографій, створення особистих цифрових фото– і відео– альбомів і т.д.

– Управління світлом у всьому будинку, світлові сцени й сценарії.

– Управління системою вентиляції й кондиціювання.

– Управління системою опалення.

– Управління сауною, басейном.

Інформаційна система:

– Установка локальної обчислювальної мережі в будинку, мережна печать, мережні ігри (можлива використання бездротових технологій).

– Вихід в Інтернет з будь-якого комп'ютера в будинку (у тому числі з мобільного).

– Віддалене управління всіма системами будинку через Інтернет.

– "Домашній офіс" з віддаленим підключенням до корпоративної мережі робочого офісу.

Система диспетчеризації:

– Система безперебійного електропостачання.

– Управління системою опалення, казаном водонагрівача.

– Контроль витоку води, газу.

– Система управління здатна погоджувати роботу інженерних систем, оцінюючи стан сенсорів, датчиків, відпрацьовуючи команди з пультів управління, прив’язуючись до часу доби, пори року й т.п. При цьому виключаються ситуації, коли домашнє устаткування, покликане вирішувати спеціалізовані проблеми, працює в режимах, взаємовиключаючих один одного.

КПІЗ_2025

3.3 Розробка функціональної схеми

Функціональна структура розробленої мережі складається з наступних блоків, як взаємодіють з блоком функцій LPWAN:

– блок функцій, які відповідають за безпеку, що дозволяє в автоматичному режимі знімати дані з датчиків і по мережі передавати їх до охоронного сервера, що приймає рішення, яку функцію виконати;

– блок функцій, які відповідають за ручне управління інтелектуальним домом, тобто дозволяють через розроблене програмне забезпечення за допомогою пультів, або персонального комп'ютера управляти усіма болками інтелектуального дому;

– блок функцій, які відповідають за роботу з мультимедіа, тобто дозволяють по локальній мережі всередині будинку передавати дані на різного виду кінцеві пристрої (плазмений телевізор, сенсорні панелі й т.і.);

– блок функцій, які відповідають за автоматизацію, що дозволяють незалежно від людини виконувати роботи по підтримці дому у належному стані, тобто сервер автоматизації приймає рішення.

Блок функцій, які відповідають за безпеку, включає в себе наступні функціональні блоки:

– звукова тривожна сигналізація;

– охоронна сигналізація;

– світлова тривожна сигналізація;

– тривожне оповіщення по телефону;

– світлова імітація наявності людей;

– функція «Паніка».

Блок функцій, які відповідають за ручне управління інтелектуальним домом включає в себе наступні функціональні блоки:

– управління з екрану ЕОМ;

– релейне управління;

КПІ-2025

– діммування;

– «проходні» схеми вимикання;

– бездротове управління електрикою;

– для аудіо/відео та світла один пульт ДУ;

– віддалене управління за телефоном.

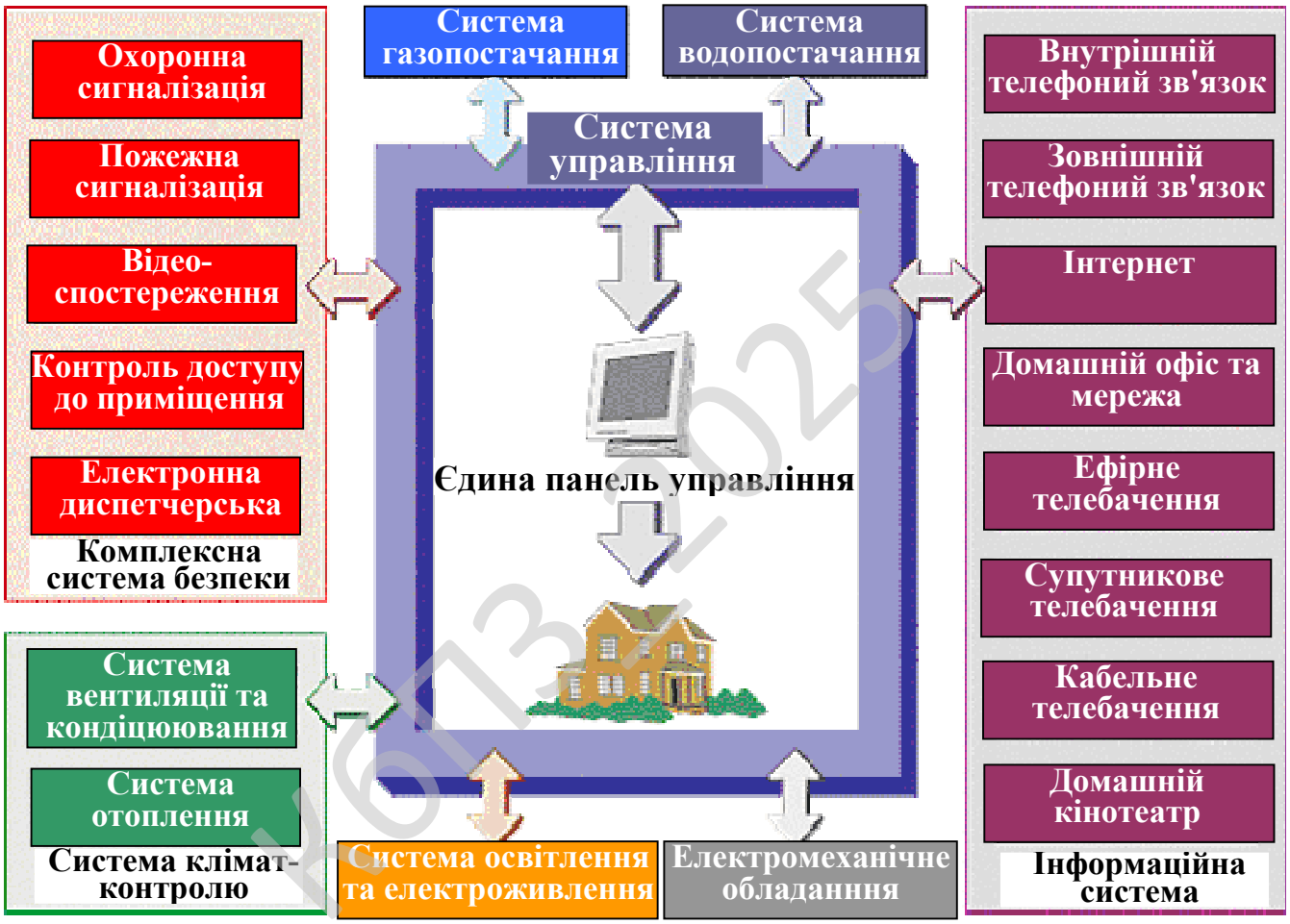

Рисунок 3.2 – Функціональна схема системи

Блок функцій, які відповідають за роботу з мультимедіа, включає в себе наступні функціональні блоки:

– бездротове аудіо/відео;

– універсальні навчаємі пульти ІЧ ДУ;

– радіотрансляція ІЧ ДУ;

– бездротове аудіо/відео спостереження;

– радіоуправління ПЗ ЕОМ.

Блок функцій, які відповідають за автоматизацію, включає в себе наступні функціональні блоки:

– макроси та сценарії;

– за температурою повітря;

– за рухом людини;

– за рівнем освітленості;

– по сухому контакту;

– за часом доби.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

КПЗ – 2025

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КПЗ-2025

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над бакалаврською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

КПЗ-2025

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

КПЗ-2025

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю IoT інтелектуального будинку з використанням мережі LPWAN.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Нижче наведемо програмний код. Який відображає підключення основних функцій у програмі.

```
//відкриття вікна "Управляння освітленням"
void __fastcall TForm_main::Button1Click(TObject *Sender)
{
Form_light->Show();
}
//---------------------------------------------------------
//відкриття вікна "Про програму..."
void __fastcall TForm_main::Button8Click(TObject *Sender)
{
Form_about->Show();
}
//---------------------------------------------------------
//відкриття вікна "Система водопостачання"
void __fastcall TForm_main::Button4Click(TObject *Sender)
{
Form_water->Show();
}
//---------------------------------------------------------
//відкриття вікна "Система клімат-контролю"
```

КбПЗ_2025

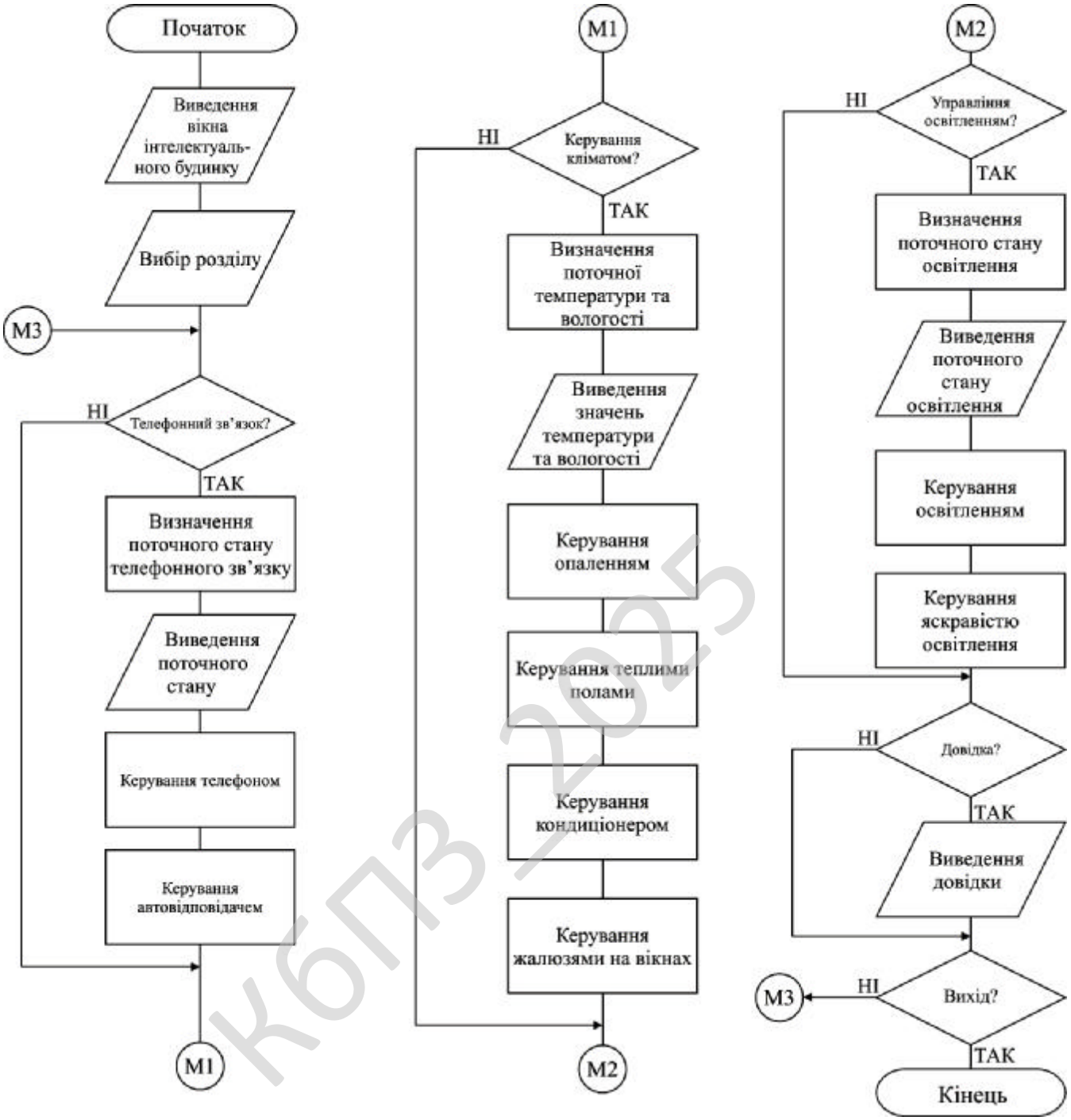

Рисунок 4.1 – Блок-схема основної програми

КОПЗ 2025

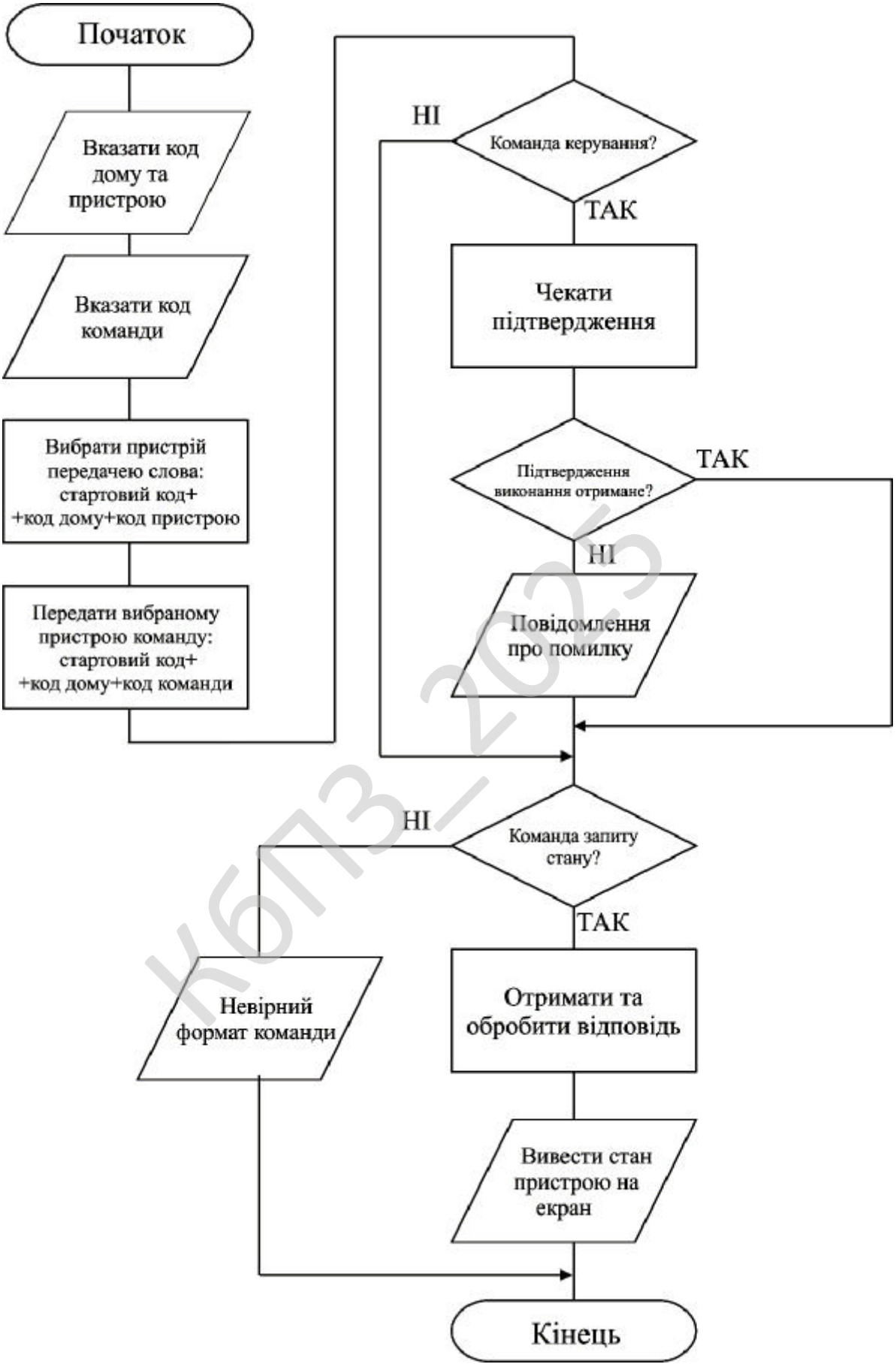

Рисунок 4.2 – Блок-схема роботи підпрограми

КБПЗ_2025

```
void __fastcall TForm_main::Button2Click(TObject *Sender)
{
Form_climate->Show();
}
//---------------------------------------------------------
//відкриття вікна "Система газопостачання"
void __fastcall TForm_main::Button5Click(TObject *Sender)
{
Form_gas->Show();        }
//---------------------------------------------------------
//відкриття вікна "Система безпеки"
void __fastcall TForm_main::Button3Click(TObject *Sender)
{
Form_security->Show();
}
//---------------------------------------------------------
//відкриття вікна "Домашній кінотеатр"
void __fastcall TForm_main::Button7Click(TObject *Sender)
{
Form_tv->Show();
}
//---------------------------------------------------------
//відкриття вікна " зв'язок"
void __fastcall TForm_main::Button6Click(TObject *Sender)
{
Form_phone->Show();
}
//---------------------------------------------------------
```

Хоча сигнали управління LPWAN передаються й приймаються безпосередньо по електричній мережі, для під'єднання пристоїв до комп'ютера можна використовувати різні топології, такі як наприклад Ethernet, RS-232, RS-485 та інші. Для вирішення поставленого у дипломному проекті завдання, було обрано інтерфейс Ethernet.

Стандарт LPWAN визначає метод і протокол передачі керуючих сигналів-команд (включити, виключити, яскравіше, темніше й т.д.) по силовій електропроводці на електронні модулі, до яких підключені керовані електропобутові й освітлювальні прилади. Усього в мережу LPWAN може бути об'єднане до 256 груп пристроїв з різними адресами.

КбПЗ_2025

З погляду логіки організації внутрімережної взаємодії всі пристрої LPWAN можна розбити на дві більші групи: контролери й виконавчі модулі.

Контролери відповідають за генерацію команд LPWAN і, крім ручного кнопкового керування, можуть мати убудований таймер або спеціалізований пристрій уведення зовнішнього впливу (датчик освітленості, фотоприймач інфрачервоного випромінювання від пульта дистанційного керування й т.д.).

Виконавчий модуль, виконуючи команди, передані тим або іншому контролеру, управляє комутацією електроживлення побутового або освітлювального приладу, відіграючи роль "розумного" вимикача. Найпоширеніші модулі двох типів: лампові (lamp module) і приладові (appliance module).

Конструктивно лампові модулі являють собою тиристорні регулятори потужності й забезпечують, крім функцій включення й вимикання, плавну регулювання яскравості світіння електроламп (функція діммера, від англійського слова dimmer – "реостат", "темнитель"). Приладові модулі оснащені електромагнітними реле для перемикання живлення й не призначені для плавного регулювання подаваної на навантаження потужності.

Для передачі команди X10 потрібно одинадцять циклів (періодів) силової напруги. Перші два цикли передають стартовий код, наступні чотири цикли представляють код будинку (з A по P) і останні п'яти циклів передають код приладу (з 1 по 16) або код функції (ВМК, ВИМК і т.д.), тобто ключовий код. Цей повний код (стартовий код + код будинку + ключовий код) завжди передається двічі безперервним блоком. Між блоками різних команд завжди повинен бути перерва в три цикли силової напруги. Виключенням із цього правила є блоки команд ЯСКРАВІШЕ/ТЕМНІШЕ, які передаються послідовно (мінімум два блоки) без затримок .

Усередині кожного блоку, код будинку й ключовий код повинні передаватися з кодами, що доповнюють до одиниці, у суміжних напівперіодах силової напруги. Наприклад, якщо одиничний імпульс переданий у першій

КОПІЯ 2025

половині періоду, то в другий не повинне бути ніякого сигналу (нульовий біт).

Нижче наведені можливі значення коду будинку й ключового коду і їхні двійкові подання. Стартовий код – це унікальний код, завжди рівний 1110 і не має суміжних напівперіодах доповнюють, що біт в, тобто значущі біти передаються на кожний перехід силової напруги через нуль.

[1] – HAIL запит (запит-вітання) передається для знаходження передавачів у зоні покриття. Це дозволяє виставити різні коди будинків у випадку одержання відповіді Hail Acknowledge.

[2] – У коді функції Pre-Set Dim, біт D8 разом із чотирма бітами коду будинку становить блок з 5 біт {D8H8H4H2H1}, що визначає абсолютний рівень діммеру.

[3] – Функція Extended Data (додаткові дані) передує послідовності байт (8 біт) довільної довжини, які представляють аналогові дані після аналогово-цифрового перетворення. Код функції й байти даних передаються безупинно, без пауз. Перший байт даних може вказувати на кількість байт у послідовності. Якщо при передачі в послідовності байт допущені паузи, то модуль – приймач може виконати помилкову операцію.

Функція Extended Code еквівалентна Extended Data: послідовність байт (без пауз), які представляють додаткові коди. Це дозволяє розроблювачам використовувати більше 256 наявних кодів.

Перші 16 із ключових кодів визначають номер модуля, що надалі буде приймати й виконувати команди (ВМК, ВИМК, ЯСКРАВІШЕ, ТЕМНІШЕ) до перевизначення керованого модуля. Біт D16 називається "функціональним бітом", якщо він дорівнює 1, то передається функція, інакше код модуля.

Приведемо приклад.

Щоб включити 5-ий модуль в "будинку К", потрібно послати по електромережі наступний рядок біт:

1110010110100101011001111001011010010101100100000011100101101001
0110011011100101101001011001 10.

КПЗ_2025

Ця посилка містить 94 біта, і займе 47 циклів силової напруги або 0,94 с (майже секунда!). Тому, коли ви натискаєте на кнопку ВМК, світло включається із запізнюванням. Реакція на команди "Все світло ВМК" або "Всі модулі ВИМК" помітно швидше, тому що не передається код модуля.

Команди LPWAN

У протоколі LPWAN передбачено шість базових команд:

1. Включити (On)
2. Виключити (Off)
3. Яскравіше (Bright)
4. Темніше (Dim)
5. Включити все світло (ALL Ligts ON)
6. Виключити всі (ALL Units OFF)

Сигнали LPWAN

Технологія LPWAN використовує цифрове подання сигналів керування. Інформація кодується двійковим кодом і передається по електричній мережі за допомогою високочастотних імпульсів. Кожний переданий імпульс відповідає одному біту інформації сj значенням “1”. Передача чергового імпульсу відбувається в момент часу, коли сіткова напруга приймає нульове значення.

Стандартна команда LPWAN передається протягом, приблизно, 50 періодів мережної напруги частоти 50Гц або 1 сек.

Більшість переданих по мережі LPWAN повідомлень містить, принаймні, два інформаційні поля: адреса пристрою, якому ця команда адресована, і властиво команду. Підключені до електромережі пристрої LPWAN приймають передані повідомлення, декодують поле адреси й, якщо він збігається з їх власним, виконують команду.

Інтеграція LPWAN

Основним засобом інтеграції мережі LPWAN із зовнішнім устаткуванням є PLC інтерфейс XM10. Будь-який контролер, що підтримує відкритий протокол обміну з XM10, може відправляти команди LPWAN в електромережу й, навпаки,

КПЗ_2025

одержувати з мережі інформацію про стан пристроїв LPWAN.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об’єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

КПЗ-2025

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізовують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Також при розробці бакалаврської дипломної роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії

КБПЗ–2025

мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

– асоціації (association relationship);

– включення (include relationship);

– розширення (extend relationship);

– узагальнення (generalization relationship).

КПЗ-2025

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі

КПЗ_2025

випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з

КПЗ-2025

класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортування.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

КПЗ_2025

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

КПЗ_2025

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм PRESENT – окремий випадок SP-мережі й складається з 31 раунду. Довжина блоку становить 64 біта, а ключі підтримуються в 2 варіантах, 80- і 128-бітні. Такого рівня захисту повинно цілком вистачати для низькозахищених додатків, звичайно використовуваних для розгортання на основі тегів, а крім того, що важливіше, PRESENT багато в чому збігається своїми конструктивними особливостями з потоковими шифрами проекту estream, заточеними на ефективну реалізацію в залозі, що дозволяє нам адекватно порівнювати їх.

Кожний з 31 раундів складається з операції XOR, щоб увести ключ K_i для $1 \leq i \leq 32$, де K_{32} використовується для «відбілювання» ключа, лінійної побітової перестановки й нелінійного шару заміщення (або, попросту говорячи, збільшення стійкості шифрування. Нелінійний шар використовує роздільні 4-бітні S-блоки, які застосовуються паралельно 16 раз на кожному раунді. Шифр, описаний псевдо-кодом представлено на рисунку 4.3.

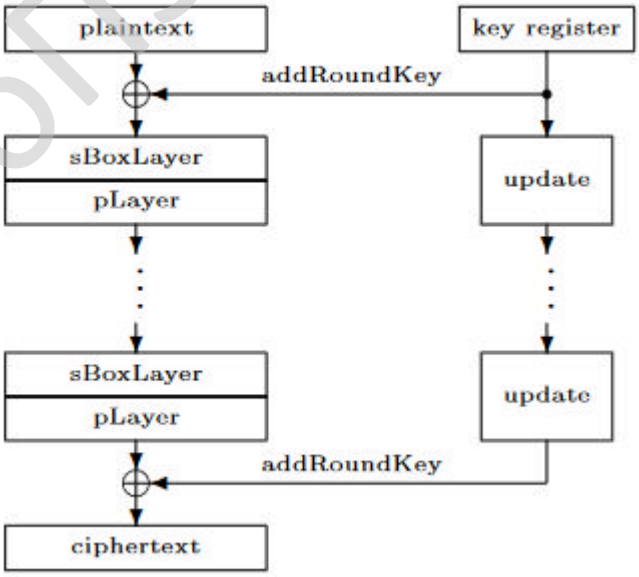

Рисунок 4.3 – Шифр PRESENT

КБПЗ_2025

Тепер кожна стадія визначається по черзі. Обґрунтування конструкції наведені нижче, а біти всюди нумеруються з нуля, починаючи із правого в блоці або слові.

Додавання раундового ключа (addRoundKey)

Заданий раундовий ключ $K_i = k^i_{63}\ldots k^i_0$, де $1 \le i \le 32$, а так само поточний стан $b_{63}\ldots b_0$. Додавання раундового ключа до поточного стану відбувається за модулем 2 ($b_j = b_j \oplus k^i_j$, де $0 \le j \le 63$).

Шар S-блоків (sBoxlayer)

Використовувані в PRESENT S-блоки відображають 4-бітні блоки в 4-бітні блоки. Дія цього блоку в шістнадцятковій системі числення наведене в наступній таблиці.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Рисунок 4.4 – Дія блоку в шістнадцятковій системі числення

Для шару S-блоків поточний стан $b_{63}\ldots b_0$ представляє із себе 16 4-бітних слів $w_{15}\ldots w_0$, де $w_i = b_{4*i+3} \,||\, b_{4*i+2} \,||\, b_{4*i+1} \,||\, b_{4*i}$ для $0 \le i \le 15$. Вихід рамки $S[w_i]$ видає оновлені значення станів очевидним образом.

Шар перестановки (player)

Побітова перестановка, використовувана в PRESENT задається наступною таблицею (біт і стану зміщається на позицію P(i).

КОПЗ_2025

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Рисунок 4.5 – Побітова перестановка, використовувана в PRESENT

Перетворення ключа (The key schedule)

PRESENT може використовувати 80- і 128-бітні ключі, однак зосередимося на 80-бітній версії. Надаваний користувачем ключ зберігається в регістрі ключів K, представлений у вигляді $k_{79}k_{78}\ldots k_0$. На i-ом раунді 64-бітний раундовий ключ $Ki = k_{63}k_{62}\ldots k_0$, що полягає з 64 лівих битов поточного вмісту регістру K. Таким чином, на i-ом раунді маємо:

$K_i = k_{63}k_{62}\ldots k_0 = k_{79}k_{78}\ldots k_{16}$.

Після розпакування раундового ключа K_i регістр ключа $K = k_{79}k_{78}\ldots k_0$ обновляється в такий спосіб:

1. $[k_{79}k_{78}\ldots k_1k_0] = [k_{18}k_{17}\ldots k_{20}k_{19}]$
2. $[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$
3. $[k_{19}k_{18}k_{17}k_{16}k_{15}] = [k_{19}k_{18}k_{17}k_{16}k_{15}] \oplus$ round_counter

Отже, регістр ключа зрушується на 61 позицію вліво, 4 крайніх лівих біта, що пройшли через S-блок і round_counter значення і складається за модулем 2 з бітами $k_{19}k_{18}k_{17}k_{16}k_{15}$ з K з найменшим значущим бітом з round_counter праворуч.

КПЗ_2025

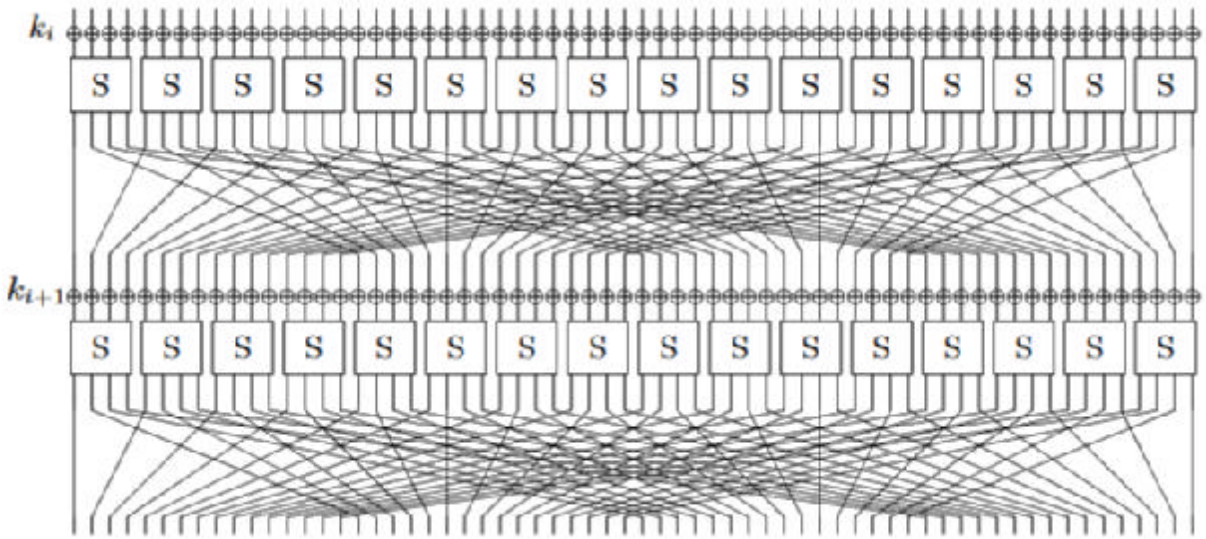

Рисунок 4.6 – Перетворення ключа (The key schedule)

Перетворення ключа для 128-бітного алгоритму можна знайти в додатку до оригінальної статті.

Конструктивні особливості PRESENT

Крім безпеки й ефективної реалізації, основне досягнення PRESENT – його простота. по цьому не дивно, що схожі проекти були прийняті в інших обставинах, і навіть були використані як навчальний посібник для студентів. У даній секції ми обґрунтуємо рішення, прийняті нами при проектуванні PRESENT. Однак, у першу чергу, опишемо очікувані прикладні вимоги.

Цілі й середовище застосування

При проектуванні блокового шифру, застосовного в жорстко обмежених оточеннях, важливо зрозуміти, що ми не створюємо блоковий шифр, який, неодмінно, буде застосовний у багатьох ситуаціях – для цього існує AES. Навпаки, ми націлені на досить специфічне застосування, для якого AES не підходить. Вищесказане визначає нам наступні характеристики:

– Шифр буде реалізований «у залізі»

КОПЗ_2025

– Додатки будуть вимагатися лише для регулювання рівня безпеки. Отже, 80-бітний ключ буде здоровим розв'язком. Відзначимо, що такої ж позиції дотримуються розроблювачі потокових шифрів проекту eSTREAM.

– Додатки не припускають шифровки великої кількості даних. Таким чином, реалізація може бути оптимізована для продуктивності або простору без внесення занадто великих змін.

– У деяких застосуваннях можлива ситуація, що ключ буде зафіксований при виробництві. У такому випадку, не треба буде змінювати ключ пристрою (що може вилитися в атаки з маніпуляцією ключем.

– Фізичний обсяг пристрою буде першим пріоритетом, після безпеки, що спричинить обмеження на пікові й середні споживання енергії, і, отже, зрушить швидкодія в область низькопріоритетних параметрів.

– У пристроях, що вимагають найбільш ефективного використання фізичного простору, блоковий шифр найчастіше зможе лише шифрувати дані (encryption-only mode). Таким чином, він зможе бути використаний у запит-відповідь (challenge-response) протоколах авторизації, і, при дотриманні контролю стану, може бути використаний для шифровки й дешифрування переговорів із пристроєм, використовуючи режим лічильника.

Виходячи з таких міркувань, розв'язали створити PRESENT як 64-бітний блоковий шифр із 80-бітним ключем. Шифровка й дешифрування, у цьому випадку, мають приблизно схожі фізичні вимоги. Маючи можливість підтримувати як шифрацію, так і дешифрацію, PRESENT буде компактніше, чим підтримуючий лише шифрацію AES. А у випадку encryption-only виконання, наш шифр виявиться й зовсім понад-легко. Суб-ключі що шифрують будуть обчислюватися на ходу.

У літературі є безліч прикладів атак компромісу між часом, датою й пам'яттю, або атак з використанням парадокса днів народження при шифровці великих обсягів даних. Однак, дані атаки залежать тільки від параметрів шифру й не використовують внутрішню структуру. Наша мета полягає в тому, щоб ці

КОПЗ-2025

атаки були кращим, що можуть застосувати проти нас. Атаки стороннього каналу й атаки з безпосереднім зломом чипа загрожують PRESENT тією самою мірою, як і іншим криптографічним примітивам. Однак для ймовірних застосувань, помірні вимоги безпеки роблять вигоду, одержувану зловмисником на практиці, досить обмеженої. В оцінці ризиків, подібні погрози не сприймаються як істотний фактор.

Перестановочний шар

При виборі шару змішування ключа, наша увага до апаратної ефективності вимагає наявності лінійного шару, який може бути реалізований з мінімальною кількістю керуючих елементів (наприклад, транзисторів. це приводить до побітової перестановки. Приділяючи увагу простоті, ми вибрали регулярну бітову перестановку, що допомагає провести прозорий аналіз безпеки.

КБПЗ_2025

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ IoT інтелектуального будинку з використанням мережі LPWAN яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційне меню: Дані; Підсистеми; Налаштування мережі; Допомога.

– Функції представлені у графічному вигляді (іконки).

– Розділу виведення результату роботи системи.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

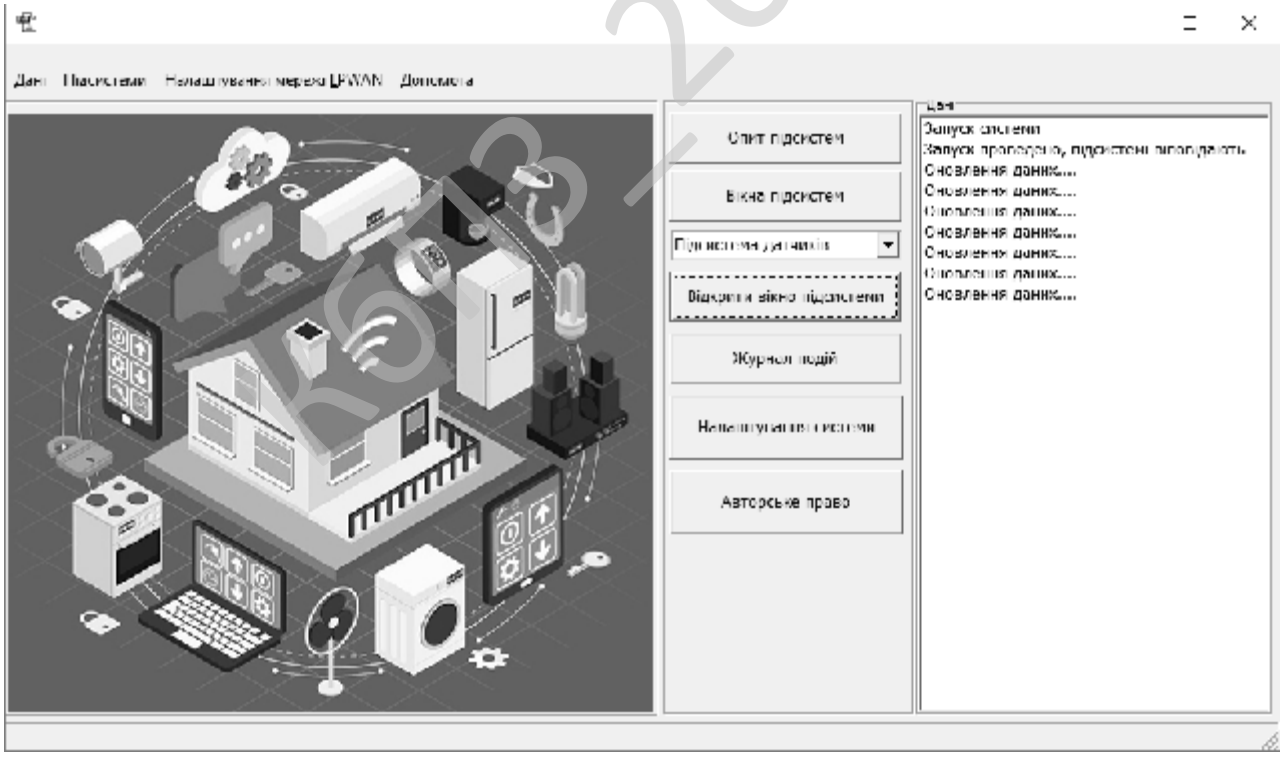

Рисунок 5.1 – Головне вікно ПЗ

КБПЗ – 2025

Областю застосування ПЗ є інтелектуальний будинок. На сьогоднішній період не існує строгого визначення поняття «розумний будинок» або «інтелектуальний будинок». Поки що кожний користувач самостійно визначає основні функції й інженерні системи у квартирі й будинку, на яких можуть бути реалізовані ці функції шляхом додаванням інтелекту керування.

І кожна така система підвищує рівень інтелектуальності будинку надаючи новий рівень безпеки, функціональності й комфорту. Головна роль у реалізації цих проектів належить компаніям, що здійснюють проектування й установку системи розумний будинок, при спільній участі власника або хазяїна зазначеного будинку.

Система розумний будинок включає, але не обмежує, що випливають функції: Дистанційний контроль роботи інженерних систем будинку – опалення, електропостачання, газової служби, водопостачання й каналізації, охоронної системи. А також систему дистанційного керування. Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку

КОПЗ-2025

виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

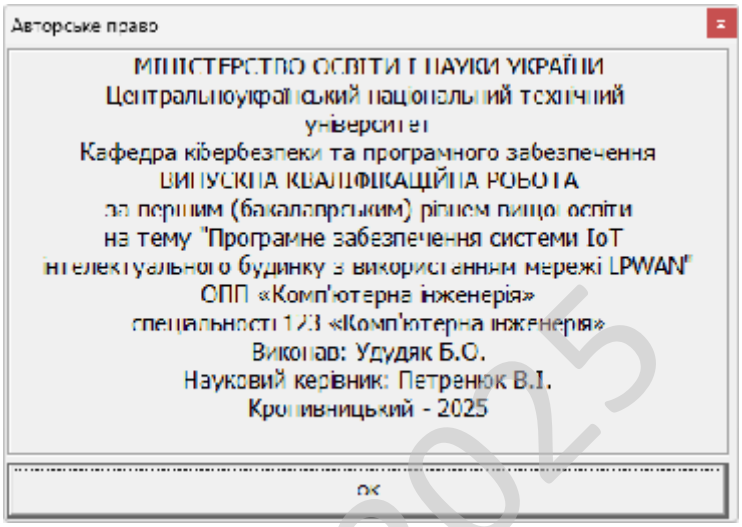

Рисунок 5.2 – Авторське право

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

КбПЗ_2025

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

КПЗ-2025

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^10. Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ITc).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його OT).

КПЗ-2025

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору OT.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій.

– Помилок інтерфейсу.

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних.

– Помилок характеристик (необхідна ємність пам'яті і т.д.).

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

КБПЗ–2025

Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ).

Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (зареєструватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КбПЗ_2025

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи IoT інтелектуального будинку з використанням мережі LPWAN.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем IoT інтелектуального будинку з використанням мережі LPWAN.

– Досліджена система IoT інтелектуального будинку з використанням мережі LPWAN.

– На основі отриманих результатів досліджень створена програмна реалізація системи IoT інтелектуального будинку з використанням мережі LPWAN.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання IoT інтелектуального будинку з використанням мережі LPWAN.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Builder C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи IoT інтелектуального будинку з використанням мережі LPWAN. Це

КОПЗ_2025

дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм PRESENT.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

КбПЗ_2025

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

2. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.

3. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.

4. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.

5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

6. Kuznetsov, O., Kuznetsova, Y., Smirnov, O., Kostenko, O., Zvieriev, V. «Evaluating Hashing Algorithms in the Age of ASIC Resistance». *CEUR Workshop Proceedings*, 2023, 3628, pp. 93-105.

7. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

8. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

9. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of

КПІ – 2025

Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.

10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

12. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

13. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

14. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

15. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

16. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

КОПЗ-2025

17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

18. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

20. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T),* Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

21. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T),* Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

22. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

23. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019*/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

КОПЗ_2025

24. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

25. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

26. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 618-629.

27. Smirnov A.A., Kuzne tsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

28. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Конопліцька-Слободенюк , О., Смірнова , Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об'єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 6-26.

29. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

30. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки

КБПЗ_2025

програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

31. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

32. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи,* 2023, том 7, № 2, С. 49-56.

33. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

34. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

35. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

36. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

КОПЗ-2025

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

37. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

38. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.

39. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

40. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

41. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія.* – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

42. Смірнов О.А., Дрєєва Г.М., Дрєєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки.* № 2(33). с. 161-172, 2019.

43. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

КОПЗ_2025

44. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

45. Смірнов О.А., Дрєєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

46. Дрєєва Г.М., Смірнов О.А., Дрєєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

47. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дрєєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

48. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Конопліцька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

49. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

50. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

КБПЗ-2025

Додаток А
(обов’язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....2
2 Підстава для розробки.....2
3 Мета та призначення розробки.....2
4 Джерела розробки.....2
5 Технічні вимоги.....2
5.1 Вміст проекту.....2
5.2 Показники призначення.....3
5.3 Вимоги до функціональних характеристик.....3
5.4 Вимоги до архітектури.....3
5.5 Вимоги до надійності.....3
5.6 Умови експлуатації.....4
5.7 Вимоги до складу та параметрів технічних засобів.....4
5.8 Вимоги до інформаційної і програмної сумісності.....4
5.8.1 Обладнання.....4
5.8.2 Мова програмування.....4
5.8.3 Вхідні дані.....5
5.8.4 Вихідні дані.....5
6 Вимоги до програмної документації.....5
7 Перелік документів, що розробляються.....5
8 Етапи розробки.....6
9 Порядок контролю та приймання.....6

КОПЗ–2025

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи IoT інтелектуального будинку з використанням мережі LPWAN.

2 Підстава для розробки

Підставою для розробки служить завдання на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 47-02 від 17.01.2025 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

КБПЗ_2025

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи IoT інтелектуального будинку з використанням мережі LPWAN;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

КБПЗ_2025

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпечення.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

КбПЗ – 2025

5.8.1 Обладнання

Комп’ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Builder C++.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 76 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

КБПЗ_2025

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2025 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 3.06.2025 р.

КбПЗ_2025

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

_____ Петренюк В.І.

Програмне забезпечення системи IoT інтелектуального будинку з використанням мережі LPWAN

КбПЗ_2025

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 48

Літера: РП

Кропивницький – 2025 року

Основна програма

Файл Project_IoT_home_LPWAN.cpp основної програми

```
#include <vcl.h>
#pragma hdrstop
//---------------------------------------------------------------------------
USEFORM("main.cpp", Form_main);
USEFORM("about.cpp", Form_about);
USEFORM("light.cpp", Form_light);
USEFORM("water.cpp", Form_water);
USEFORM("gas.cpp", Form_gas);
USEFORM("security.cpp", Form_security);
USEFORM("climate.cpp", Form_climate);
USEFORM("phone.cpp", Form_phone);
USEFORM("tv.cpp", Form_tv);
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
        try
        {
                 Application->Initialize();
                 Application->CreateForm(__classid(TForm_main), &Form_main);
                 Application->CreateForm(__classid(TForm_about), &Form_about);
                 Application->CreateForm(__classid(TForm_light), &Form_light);
                 Application->CreateForm(__classid(TForm_water), &Form_water);
                 Application->CreateForm(__classid(TForm_gas), &Form_gas);
                 Application->CreateForm(__classid(TForm_security),
&Form_security);
                 Application->CreateForm(__classid(TForm_climate),
&Form_climate);
                 Application->CreateForm(__classid(TForm_phone), &Form_phone);
                 Application->CreateForm(__classid(TForm_tv), &Form_tv);

                 Application->Run();
        }
        catch (Exception &exception)
        {
                 Application->ShowException(&exception);
        }
        catch (...)
        {
                 try
                 {
                         throw Exception("");
                 }
                 catch (Exception &exception)
                 {
                         Application->ShowException(&exception);
                 }
        }
        return 0;
}
//---------------------------------------------------------------------------
```

КОПЗ_2015

Файл main.cpp основної програми

```
//---------------------------------------------------------------------
//підключення бібліотек
#include <vcl.h>
#pragma hdrstop

//підключення модулів програми
#include "main.h"
#include "light.h"
#include "water.h"
#include "gas.h"
#include "security.h"
#include "climate.h"
#include "phone.h"
#include "tv.h"
#include "about.h"

//---------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_main *Form_main;
//---------------------------------------------------------------------
__fastcall TForm_main::TForm_main(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------

//відкриття вікна "Управляння освітленням"
void __fastcall TForm_main::Button1Click(TObject *Sender)
{
Form_light->Show();
}
//---------------------------------------------------------------------


//відкриття вікна "Про програму..."
void __fastcall TForm_main::Button8Click(TObject *Sender)
{
Form_about->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Система водопостачання"
void __fastcall TForm_main::Button4Click(TObject *Sender)
{
Form_water->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Система клімат-контролю"
void __fastcall TForm_main::Button2Click(TObject *Sender)
{
Form_climate->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Система газопостачання"
void __fastcall TForm_main::Button5Click(TObject *Sender)
{
Form_gas->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Система безпеки"
```

КОПЗ_2025

```
void __fastcall TForm_main::Button3Click(TObject *Sender)
{
Form_security->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Домашній кінотеатр"
void __fastcall TForm_main::Button7Click(TObject *Sender)
{
Form_tv->Show();
}
//---------------------------------------------------------------------

//відкриття вікна "Телефонній зв'язок"
void __fastcall TForm_main::Button6Click(TObject *Sender)
{
Form_phone->Show();
}
//---------------------------------------------------------------------
```

КбПЗ_2025

Файл main.h – бібліотека для файлу main.cpp

```
//---------------------------------------------------------------------
#ifndef mainH
#define mainH
//---------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
#include <Buttons.hpp>
//---------------------------------------------------------------------
class TForm_main : public TForm
{
__published:      // IDE-managed Components
        TImage *Image1;
        TButton *Button1;
        TButton *Button2;
        TButton *Button3;
        TButton *Button4;
        TButton *Button5;
        TButton *Button6;
        TButton *Button7;
        TImage *Image2;
        TImage *Image3;
        TImage *Image4;
        TImage *Image5;
        TImage *Image6;
        TImage *Image7;
        TImage *Image8;
        TButton *Button8;
        TImage *Image9;
        TLabel *Label1;
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button8Click(TObject *Sender);
        void __fastcall Button4Click(TObject *Sender);
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button5Click(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall Button7Click(TObject *Sender);
        void __fastcall Button6Click(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_main(TComponent* Owner);
};
//---------------------------------------------------------------------
extern PACKAGE TForm_main *Form_main;
//---------------------------------------------------------------------
#endif
```

КОПЗ_2025

Файл light.cpp – керування освітленням

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "light.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_light *Form_light;
//---------------------------------------------------------------------------
__fastcall TForm_light::TForm_light(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

//ввімкнення ліхтаря біля входу в будинок з вказаною яскравістю
void __fastcall TForm_light::torch_onClick(TObject *Sender)
{
torch->Caption="Ввімкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
TrackBar_torch->Enabled=true;

ClientSocket->Send_LPWANCommand(0,1,18,1);  //0-код будинку; 1-код пристрою, що
керує ліхтарем; 18-код команди "on"; 1-кількість повторних надсилань команди
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position); //встановлення
яскравості, вказаної користувачем за допомогою TrackBar-у


}
//---------------------------------------------------------------------------
//вимкнення ліхтаря біля входу в будинок
void __fastcall TForm_light::torch_offClick(TObject *Sender)
{
torch->Caption="Вимкнено";
Image_torch_off->Visible=true;
Image_torch_on->Visible=false;
TrackBar_torch->Enabled=false;
ClientSocket->Send_LPWANCommand(0,1,19,1);  //0-код будинку; 1-код пристрою, що
керує ліхтарем; 19-код команди "off"; 1-кількість повторних надсилань команди

}
//---------------------------------------------------------------------------
//ввімкнення світла в коридорі з вказаною яскравістю
void __fastcall TForm_light::corridor_onClick(TObject *Sender)
{
corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
TrackBar_corridor->Enabled=true;

ClientSocket->Send_LPWANCommand(0,2,18,1);
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);

}
//---------------------------------------------------------------------------

//ввімкнення світла у вітальні з вказаною яскравістю
void __fastcall TForm_light::drawing_room_onClick(TObject *Sender)
{
drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
```

КПЗ_2025

```
Image_drawing_room_off->Visible=false;
TrackBar_drawing_room->Enabled=true;

ClientSocket->Send_LPWANCommand(0,3,18,1);
ClientSocket->SendLeviton_LPWAN(0,3,TrackBar_drawing_room->Position);
}
//---------------------------------------------------------------------------

//ввімкнення світла на кухні з вказаною яскравістю
void __fastcall TForm_light::kitchen_onClick(TObject *Sender)
{
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;

ClientSocket->Send_LPWANCommand(0,4,18,1);
ClientSocket->SendLeviton_LPWAN(0,4,TrackBar_kitchen->Position);
}
//---------------------------------------------------------------------------

//ввімкнення світла в кабінеті з вказаною яскравістю
void __fastcall TForm_light::cabinet_onClick(TObject *Sender)
{
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
TrackBar_cabinet->Enabled=true;

ClientSocket->Send_LPWANCommand(0,5,18,1);
ClientSocket->SendLeviton_LPWAN(0,5,TrackBar_cabinet->Position);
}
//---------------------------------------------------------------------------

//ввімкнення світла у спальні з вказаною яскравістю
void __fastcall TForm_light::bedroom_onClick(TObject *Sender)
{
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
TrackBar_bedroom->Enabled=true;

ClientSocket->Send_LPWANCommand(0,6,18,1);
ClientSocket->SendLeviton_LPWAN(0,6,TrackBar_bedroom->Position);
}
//---------------------------------------------------------------------------

//ввімкнення світла в дитячій кімнаті з вказаною яскравістю
void __fastcall TForm_light::baby_room_onClick(TObject *Sender)
{
baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
TrackBar_baby_room->Enabled=true;

ClientSocket->Send_LPWANCommand(0,7,18,1);
ClientSocket->SendLeviton_LPWAN(0,7,TrackBar_baby_room->Position);
}
//---------------------------------------------------------------------------

//ввімкнення світла у ванній кімнаті з вказаною яскравістю
void __fastcall TForm_light::bathroom_onClick(TObject *Sender)
{
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;
ClientSocket->Send_LPWANCommand(0,8,18,1);
ClientSocket->SendLeviton_LPWAN(0,8,TrackBar_bathroom->Position);
```

КОПЗ_2025

```
}

//---------------------------------------------------------------------------

//вимкнення світла в коридорі
void __fastcall TForm_light::corridor_offClick(TObject *Sender)
{
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
TrackBar_corridor->Enabled=false;
ClientSocket->Send_LPWANCommand(0,2,19,1);
}
//---------------------------------------------------------------------------

//вимкнення світла у вітальні
void __fastcall TForm_light::drawing_room_offClick(TObject *Sender)
{
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
TrackBar_drawing_room->Enabled=false;
ClientSocket->Send_LPWANCommand(0,3,19,1);
}
//---------------------------------------------------------------------------

//вимкнення світла на кухні
void __fastcall TForm_light::kitchen_offClick(TObject *Sender)
{
kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
TrackBar_kitchen->Enabled=false;
ClientSocket->Send_LPWANCommand(0,4,19,1);
}
//---------------------------------------------------------------------------

//вимкнення світла в кабінеті
void __fastcall TForm_light::cabinet_offClick(TObject *Sender)
{
cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
TrackBar_cabinet->Enabled=false;
ClientSocket->Send_LPWANCommand(0,5,19,1);
}
//---------------------------------------------------------------------------

//вимкнення світла у спальні
void __fastcall TForm_light::bedroom_offClick(TObject *Sender)
{
bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
TrackBar_bedroom->Enabled=false;
ClientSocket->Send_LPWANCommand(0,6,19,1);
}
//---------------------------------------------------------------------------

//вимкнення світла у дитячій кімнаті
void __fastcall TForm_light::baby_room_offClick(TObject *Sender)
{
baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
TrackBar_baby_room->Enabled=false;
ClientSocket->Send_LPWANCommand(0,7,19,1);
}
```

КПЗ_2025

```
//---------------------------------------------------------------------------

//вимкнення світла у ванній кімнаті
void __fastcall TForm_light::bathroom_offClick(TObject *Sender)
{
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
TrackBar_bathroom->Enabled=false;
ClientSocket->Send_LPWANCommand(0,8,19,1);
}
//---------------------------------------------------------------------------

//ввімкнення світла скрізь
void __fastcall TForm_light::Button18Click(TObject *Sender)
{

torch->Caption="Ввімкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,1,18,1);
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position);

corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,2,18,1);
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);

drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,3,18,1);
ClientSocket->SendLeviton_LPWAN(0,3,TrackBar_drawing_room->Position);

kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,4,18,1);
ClientSocket->SendLeviton_LPWAN(0,4,TrackBar_kitchen->Position);

cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,5,18,1);
ClientSocket->SendLeviton_LPWAN(0,5,TrackBar_cabinet->Position);

bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,6,18,1);
ClientSocket->SendLeviton_LPWAN(0,6,TrackBar_bedroom->Position);

baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,7,18,1);
ClientSocket->SendLeviton_LPWAN(0,7,TrackBar_baby_room->Position);


bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
ClientSocket->Send_LPWANCommand(0,8,18,1);
ClientSocket->SendLeviton_LPWAN(0,8,TrackBar_bathroom->Position);


TrackBar_torch->Enabled=true;
TrackBar_bedroom->Enabled=true;
```

КПІ – 2025

```
TrackBar_corridor->Enabled=true;
TrackBar_drawing_room->Enabled=true;
TrackBar_kitchen->Enabled=true;
TrackBar_cabinet->Enabled=true;
TrackBar_baby_room->Enabled=true;
TrackBar_bathroom->Enabled=true;
}
//---------------------------------------------------------------------------

//вимкнення світла скрізь
void __fastcall TForm_light::Button17Click(TObject *Sender)
{
torch->Caption="Вимкнено";
Image_torch_off->Visible=true;
Image_torch_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,1,19,1);

corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,2,19,1);

drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,3,19,1);

kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,4,19,1);

cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,5,19,1);

bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,6,19,1);

baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,7,19,1);


bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
ClientSocket->Send_LPWANCommand(0,8,19,1);

TrackBar_torch->Enabled=false;
TrackBar_bedroom->Enabled=false;
TrackBar_corridor->Enabled=false;
TrackBar_drawing_room->Enabled=false;
TrackBar_kitchen->Enabled=false;
TrackBar_cabinet->Enabled=false;
TrackBar_baby_room->Enabled=false;
TrackBar_bathroom->Enabled=false;
}
//---------------------------------------------------------------------------

//імітація присутності господарів
//ввімкнення та вимкнення світла випадковим чином
void __fastcall TForm_light::Button1Click(TObject *Sender)
{
```

КОПІЯ 2025

```
if(Button1->Caption=="Імітація присутності")
{
Timer1->Enabled=true;      //затуск таймеру, що запрограмований на імітацію
присутності
Button1->Caption=="Вимкнути імітацію"
}
else
{
Timer1->Enabled=false;     //зупинтка таймеру
Button1->Caption=="Імітація присутності"
}

}
//---------------------------------------------------------------------------

//таймер, запрограмований на імітацію присутності
void __fastcall TForm_light::Timer1Timer(TObject *Sender)
{
int x, y;
randomize();
x=random (6)+2;  //генерація випадкового номера лампи в діапазоні 2-8
ClientSocket->Send_LPWANCommand(0,x,18,1);
y=random (6)+2;
ClientSocket->Send_LPWANCommand(0,y,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_light::brightnessTimer(TObject *Sender)
{
//зміна яскравості
}
//---------------------------------------------------------------------------

//зміна яскравості ліхтаря
void __fastcall TForm_light::TrackBar_torchChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,1,TrackBar_torch->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення коридору
void __fastcall TForm_light::TrackBar_corridorChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_corridor->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення вітальні
void __fastcall TForm_light::TrackBar_drawing_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_drawing_room->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення кухні
void __fastcall TForm_light::TrackBar_kitchenChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_kitchen->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення кабінету
void __fastcall TForm_light::TrackBar_cabinetChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_cabinet->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення спальні
```

КПЗ-2025

```
void __fastcall TForm_light::TrackBar_bedroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_bedroom->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення дитячої
void __fastcall TForm_light::TrackBar_baby_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_baby_room->Position);
}
//---------------------------------------------------------------------------

//зміна яскравості освітлення ванної
void __fastcall TForm_light::TrackBar_bathroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,2,TrackBar_bathroom->Position);
}
//---------------------------------------------------------------------------

//визначення та виведення на екран поточного стану освітлення
void __fastcall TForm_light::StatusTimer(TObject *Sender)
{
ShortString st;

ClientSocket->Send_LPWANStatusQuery();

st=_LPWANState->GetStateOnOff(0, 1);
if(st="On") {
torch->Caption="Ввімкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
TrackBar_torch->Enabled=true;
}
else {
torch->Caption="Вимкнено";
Image_torch_off->Visible=true;
Image_torch_on->Visible=false;
TrackBar_torch->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 2);
if(st="On") {
corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
TrackBar_corridor->Enabled=true;
}
else {
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
TrackBar_corridor->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 3);
if(st="On") {
drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
TrackBar_drawing_room->Enabled=true;
}
else {
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
TrackBar_drawing_room->Enabled=false;
}
```

КПІ_2025

```
st=_LPWANState->GetStateOnOff(0, 4);
if(st="On") {
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;
}
else {
kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
TrackBar_kitchen->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 5);
if(st="On") {
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
TrackBar_cabinet->Enabled=true;
}
else {
cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
TrackBar_cabinet->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 6);
if(st="On") {
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
TrackBar_bedroom->Enabled=true;
}
else {
bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
TrackBar_bedroom->Enabled=false;
}

st=_LPWANState->GetStateOnOff(0, 7);
if(st="On") {
baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
TrackBar_baby_room->Enabled=true;
}
else {
baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
TrackBar_baby_room->Enabled=false;
}
st=_LPWANState->GetStateOnOff(0, 8);
if(st="On") {
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;
}
else {
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
TrackBar_bathroom->Enabled=false;
}
}
```

КбТЗ_2025

Файл light.h – бібліотека для файлу light.cpp

```
#ifndef lightH
#define lightH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//---------------------------------------------------------------------------
class TForm_light : public TForm
{
__published:      // IDE-managed Components
        TTrackBar *TrackBar_corridor;
        TLabel *Label11;
        TImage *Image_torch_on;
        TImage *Image_torch_off;
        TTrackBar *TrackBar_torch;
        TLabel *Label8;
        TImage *Image_corridor_on;
        TImage *Image_corridor_off;
        TTrackBar *TrackBar_drawing_room;
        TLabel *Label9;
        TImage *Image_drawing_room_on;
        TImage *Image_drawing_room_off;
        TTrackBar *TrackBar_cabinet;
        TLabel *Label12;
        TImage *Image_cabinet_on;
        TImage *Image_cabinet_off;
        TTrackBar *TrackBar_bedroom;
        TLabel *Label13;
        TImage *Image_bedroom_on;
        TImage *Image_bedroom_off;
        TTrackBar *TrackBar_baby_room;
        TLabel *Label14;
        TImage *Image_baby_room_on;
        TImage *Image_baby_room_off;
        TTrackBar *TrackBar_kitchen;
        TLabel *Label15;
        TImage *Image_kitchen_on;
        TImage *Image_kitchen_off;
        TTrackBar *TrackBar_bathroom;
        TLabel *Label16;
        TImage *Image_bathroom_on;
        TImage *Image_bathroom_off;
        TButton *Button17;
        TButton *Button18;
        TLabel *Label17;
        TLabel *torch;
        TLabel *Label19;
        TLabel *corridor;
        TLabel *Label21;
        TLabel *drawing_room;
        TLabel *Label23;
        TLabel *kitchen;
        TLabel *Label25;
        TLabel *cabinet;
        TLabel *Label27;
        TLabel *bedroom;
        TLabel *Label29;
        TLabel *baby_room;
        TLabel *Label31;
        TLabel *bathroom;
        TBevel *Bevel1;
```

КБТЭ_2025

```
        TLabel *Label10;
        TBevel *Bevel2;
        TLabel *Label1;
        TBevel *Bevel3;
        TBevel *Bevel4;
        TBevel *Bevel5;
        TBevel *Bevel6;
        TBevel *Bevel7;
        TBevel *Bevel8;
        TLabel *Label3;
        TLabel *Label4;
        TLabel *Label7;
        TLabel *Label2;
        TLabel *Label6;
        TLabel *Label5;
        TButton *torch_on;
        TButton *torch_off;
        TButton *corridor_on;
        TButton *corridor_off;
        TButton *drawing_room_on;
        TButton *drawing_room_off;
        TButton *kitchen_on;
        TButton *kitchen_off;
        TButton *cabinet_on;
        TButton *cabinet_off;
        TButton *bedroom_on;
        TButton *bedroom_off;
        TButton *baby_room_on;
        TButton *baby_room_off;
        TButton *bathroom_on;
        TButton *bathroom_off;
        TButton *Button1;
        TTimer *Timer1;
        TTimer *Status;
        void __fastcall torch_onClick(TObject *Sender);
        void __fastcall torch_offClick(TObject *Sender);
        void __fastcall corridor_onClick(TObject *Sender);
        void __fastcall drawing_room_onClick(TObject *Sender);
        void __fastcall kitchen_onClick(TObject *Sender);
        void __fastcall cabinet_onClick(TObject *Sender);
        void __fastcall bedroom_onClick(TObject *Sender);
        void __fastcall baby_room_onClick(TObject *Sender);
        void __fastcall bathroom_onClick(TObject *Sender);
        void __fastcall corridor_offClick(TObject *Sender);
        void __fastcall drawing_room_offClick(TObject *Sender);
        void __fastcall kitchen_offClick(TObject *Sender);
        void __fastcall cabinet_offClick(TObject *Sender);
        void __fastcall bedroom_offClick(TObject *Sender);
        void __fastcall baby_room_offClick(TObject *Sender);
        void __fastcall bathroom_offClick(TObject *Sender);
        void __fastcall Button18Click(TObject *Sender);
        void __fastcall Button17Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Timer1Timer(TObject *Sender);
        void __fastcall brightnessTimer(TObject *Sender);
        void __fastcall TrackBar_torchChange(TObject *Sender);
        void __fastcall TrackBar_corridorChange(TObject *Sender);
        void __fastcall TrackBar_drawing_roomChange(TObject *Sender);
        void __fastcall TrackBar_kitchenChange(TObject *Sender);
        void __fastcall TrackBar_cabinetChange(TObject *Sender);
        void __fastcall TrackBar_bedroomChange(TObject *Sender);
        void __fastcall TrackBar_baby_roomChange(TObject *Sender);
        void __fastcall TrackBar_bathroomChange(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_light(TComponent* Owner);
};
//---------------------------------------------------------------------------
```

KCT3 2025

```
extern PACKAGE TForm_light *Form_light;
//---------------------------------------------------------------------------
#endif
```

КбПЗ_2025

Файл climate.cpp – керування кліматом

```
#include <vcl.h>
#pragma hdrstop

#include "climate.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_climate *Form_climate;
//---------------------------------------------------------------------------
__fastcall TForm_climate::TForm_climate(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm_climate::Button3Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,21,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button4Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,21,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button1Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,22,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button2Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,22,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button5Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,23,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button6Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,23,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button7Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,24,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button8Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,24,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button9Click(TObject *Sender)
{
```

КОПЗ_2025

```
ClientSocket->Send_LPWANCommand(0,26,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button10Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,26,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button11Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,27,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button12Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,27,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button13Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,28,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button14Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,28,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button15Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,29,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::Button16Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,29,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::TrackBar1Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,21,TrackBar1->Position);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::TrackBar2Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,22,TrackBar2->Position);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::TrackBar3Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,23,TrackBar3->Position);
}
//---------------------------------------------------------------------------

void __fastcall TForm_climate::TrackBar4Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,24,TrackBar4->Position);
}
//---------------------------------------------------------------------------
```

КБПЗ_2025

```
void __fastcall TForm_climate::TrackBar5Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,24,TrackBar5->Position);
}
//---------------------------------------------------------------------------



//визначення та виведення на екран поточного стану клімат-контролю

void __fastcall TForm_climate::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 21);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 22);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 23);
if(st="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 24);
if(st="On") Label_4->Caption="Ввімкнено"; else Label_4->Caption="Вимкнено";

n=GetVariable(25);
Label_t->Caption=n;
n=GetVariable(34);
Label_v->Caption=n;

n=GetVariable(22);
Label_tp1->Caption=n;
n=GetVariable(23);
Label_tp2->Caption=n;

st=_LPWANState->GetStateOnOff(0, 26);
if(st="On") Label_5->Caption="Відкрито"; else Label_5->Caption="Закрито";
st=_LPWANState->GetStateOnOff(0, 27);
if(st="On") Label_6->Caption="Відкрито"; else Label_6->Caption="Закрито";
st=_LPWANState->GetStateOnOff(0, 28);
if(st="On") Label_7->Caption="Відкрито"; else Label_7->Caption="Закрито";
st=_LPWANState->GetStateOnOff(0, 29);
if(st="On") Label_8->Caption="Відкрито"; else Label_8->Caption="Закрито";
}
```

КСТВ_2025

Файл climate.h – бібліотека для файлу climate.cpp

```
#ifndef climateH
#define climateH
//---------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//---------------------------------------------------------------------
class TForm_climate : public TForm
{
__published:      // IDE-managed Components
        TGroupBox *GroupBox2;
        TLabel *Label3;
        TLabel *Label_1;
        TButton *Button3;
        TButton *Button4;
        TTrackBar *TrackBar1;
        TLabel *Label5;
        TLabel *Label6;
        TLabel *Label8;
        TLabel *Label9;
        TLabel *Label10;
        TLabel *Label11;
        TLabel *Label12;
        TLabel *Label13;
        TLabel *Label7;
        TGroupBox *GroupBox1;
        TLabel *Label1;
        TLabel *Label_2;
        TLabel *Label14;
        TLabel *Label15;
        TLabel *Label16;
        TLabel *Label17;
        TLabel *Label18;
        TLabel *Label19;
        TLabel *Label20;
        TLabel *Label21;
        TLabel *Label22;
        TLabel *Label27;
        TLabel *Label_tp1;
        TLabel *Label29;
        TLabel *Label30;
        TBevel *Bevel2;
        TButton *Button1;
        TButton *Button2;
        TTrackBar *TrackBar2;
        TGroupBox *GroupBox3;
        TLabel *Label31;
        TLabel *Label_3;
        TLabel *Label33;
        TLabel *Label34;
        TLabel *Label35;
        TLabel *Label36;
        TLabel *Label37;
        TLabel *Label38;
        TLabel *Label39;
        TLabel *Label40;
        TLabel *Label41;
        TLabel *Label42;
        TLabel *Label_tp2;
        TLabel *Label44;
        TLabel *Label45;
```

КОПЗ_2025

```
TBevel *Bevel3;
TButton *Button5;
TButton *Button6;
TTrackBar *TrackBar3;
TGroupBox *GroupBox4;
TLabel *Label46;
TLabel *Label_4;
TLabel *Label48;
TLabel *Label49;
TButton *Button7;
TButton *Button8;
TGroupBox *GroupBox5;
TLabel *Label57;
TLabel *Label_5;
TButton *Button9;
TButton *Button10;
TGroupBox *GroupBox6;
TLabel *Label59;
TLabel *Label_6;
TButton *Button11;
TButton *Button12;
TGroupBox *GroupBox7;
TLabel *Label61;
TLabel *Label_7;
TButton *Button13;
TButton *Button14;
TGroupBox *GroupBox8;
TLabel *Label63;
TLabel *Label_8;
TButton *Button15;
TButton *Button16;
TGroupBox *GroupBox9;
TLabel *Label67;
TLabel *Label68;
TLabel *Label_t;
TLabel *Label70;
TLabel *Label71;
TLabel *Label_v;
TImage *Image8;
TImage *Image7;
TTrackBar *TrackBar4;
TLabel *Label66;
TLabel *Label73;
TLabel *Label74;
TLabel *Label75;
TLabel *Label76;
TLabel *Label77;
TLabel *Label78;
TLabel *Label79;
TLabel *Label65;
TLabel *Label23;
TTrackBar *TrackBar5;
TLabel *Label24;
TLabel *Label25;
TLabel *Label26;
TLabel *Label50;
TLabel *Label51;
TLabel *Label52;
TLabel *Label53;
TLabel *Label54;
TLabel *Label55;
TLabel *Label56;
TLabel *Label80;
TLabel *Label81;
TButton *Button17;
TLabel *Label82;
TLabel *Label83;
TEdit *Edit1;
TLabel *Label84;
```

КОПЗ_2025

```
        TEdit *Edit2;
        TLabel *Label85;
        TTimer *Status;
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall Button4Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button5Click(TObject *Sender);
        void __fastcall Button6Click(TObject *Sender);
        void __fastcall Button7Click(TObject *Sender);
        void __fastcall Button8Click(TObject *Sender);
        void __fastcall Button9Click(TObject *Sender);
        void __fastcall Button10Click(TObject *Sender);
        void __fastcall Button11Click(TObject *Sender);
        void __fastcall Button12Click(TObject *Sender);
        void __fastcall Button13Click(TObject *Sender);
        void __fastcall Button14Click(TObject *Sender);
        void __fastcall Button15Click(TObject *Sender);
        void __fastcall Button16Click(TObject *Sender);
        void __fastcall TrackBar1Change(TObject *Sender);
        void __fastcall TrackBar2Change(TObject *Sender);
        void __fastcall TrackBar3Change(TObject *Sender);
        void __fastcall TrackBar4Change(TObject *Sender);
        void __fastcall TrackBar5Change(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_climate(TComponent* Owner);
};
//---------------------------------------------------------------------
extern PACKAGE TForm_climate *Form_climate;
//---------------------------------------------------------------------
#endif
```

КбПЗ_2025

Файл water.cpp – керування системою водопостачання

```
#include <vcl.h>
#pragma hdrstop

#include "water.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_water *Form_water;
//---------------------------------------------------------------------------
__fastcall TForm_water::TForm_water(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
//ввімкнення системи контролю протікання води у ванній
void __fastcall TForm_water::Button2Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,9,18,1);
}
//---------------------------------------------------------------------------

//відкриття клапану холодної води у ванні
void __fastcall TForm_water::Button6Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,10,18,1);
}
//---------------------------------------------------------------------------

//відкриття клапану гарячої води у ванні
void __fastcall TForm_water::Button8Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,11,18,1);
}
//---------------------------------------------------------------------------
//ввімкнення системи контролю протікання води на кухні
void __fastcall TForm_water::Button4Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,12,18,1);
}
//---------------------------------------------------------------------------
//відкриття клапану холодної води на кухні
void __fastcall TForm_water::Button10Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,13,18,1);
}
//---------------------------------------------------------------------------

//відкриття клапану гарячої води на кухні
void __fastcall TForm_water::Button12Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,14,18,1);
}
//---------------------------------------------------------------------------

//вимкнення системи контролю протікання води у ванній
void __fastcall TForm_water::Button1Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,9,19,1);
}
//---------------------------------------------------------------------------
//закриття клапану холодної води у ванні
void __fastcall TForm_water::Button5Click(TObject *Sender)
{
```

КОПЗ – 2025

```
ClientSocket->Send_LPWANCommand(0,10,19,1);
}
//---------------------------------------------------------------------------
//закриття клапану гарячої води у ванні
void __fastcall TForm_water::Button7Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,11,19,1);
}
//---------------------------------------------------------------------------

//вимкнення системи контролю протікання води на кухні
void __fastcall TForm_water::Button3Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,12,19,1);
}
//---------------------------------------------------------------------------
//закриття клапану холодної води на кухні
void __fastcall TForm_water::Button9Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,13,19,1);
}
//---------------------------------------------------------------------------

//закриття клапану гарячої води на кухні
void __fastcall TForm_water::Button11Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,14,19,1);
}
//визначення та виведення на екран поточного стану системи водопостачання

void __fastcall TForm_water::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 9);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 12);
if(st="On") Label_4->Caption="Ввімкнено"; else Label_4->Caption="Вимкнено";

n=GetVariable(9);
if(n="0") Label_11->Caption="В нормі"; else Label_11->Caption="Протікання води";
n=GetVariable(12);
if(n="0") Label_44->Caption="В нормі"; else Label_44->Caption="Протікання води";

st=_LPWANState->GetStateOnOff(0, 10);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 11);
if(st="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";

st=_LPWANState->GetStateOnOff(0, 13);
if(st="On") Label_5->Caption="Ввімкнено"; else Label_5->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 14);
if(st="On") Label_6->Caption="Ввімкнено"; else Label_6->Caption="Вимкнено";
}
```

КОПІЯ 2025

Файл water.h – бібліотека для файлу water.cpp

```
#ifndef waterH
#define waterH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//---------------------------------------------------------------------------
class TForm_water : public TForm
{
__published:      // IDE-managed Components
        TGroupBox *GroupBox1;
        TButton *Button1;
        TButton *Button2;
        TLabel *Label1;
        TLabel *Label_1;
        TGroupBox *GroupBox2;
        TLabel *Label3;
        TLabel *Label_4;
        TButton *Button3;
        TButton *Button4;
        TGroupBox *GroupBox3;
        TLabel *Label5;
        TLabel *Label_2;
        TButton *Button5;
        TButton *Button6;
        TGroupBox *GroupBox4;
        TLabel *Label7;
        TLabel *Label_3;
        TButton *Button7;
        TButton *Button8;
        TGroupBox *GroupBox5;
        TLabel *Label9;
        TLabel *Label_5;
        TButton *Button9;
        TButton *Button10;
        TGroupBox *GroupBox6;
        TLabel *Label11;
        TLabel *Label_6;
        TButton *Button11;
        TButton *Button12;
        TLabel *Label13;
        TLabel *Label_11;
        TLabel *Label15;
        TLabel *Label_44;
        TTimer *Status;
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button6Click(TObject *Sender);
        void __fastcall Button8Click(TObject *Sender);
        void __fastcall Button4Click(TObject *Sender);
        void __fastcall Button10Click(TObject *Sender);
        void __fastcall Button12Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button5Click(TObject *Sender);
        void __fastcall Button7Click(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall Button9Click(TObject *Sender);
        void __fastcall Button11Click(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_water(TComponent* Owner);
};
```

КПІЗ_2025

```
//---------------------------------------------------------------------------
extern PACKAGE TForm_water *Form_water;
//---------------------------------------------------------------------------
#endif
```

КбПЗ_2025

Файл gas.cpp – керування системою газопостачання

```
#include <vcl.h>
#pragma hdrstop

#include "gas.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_gas *Form_gas;
//---------------------------------------------------------------------------
__fastcall TForm_gas::TForm_gas(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm_gas::Button2Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,15,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_gas::Button1Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,15,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_gas::Button4Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,16,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_gas::Button3Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,16,19,1);
}
//---------------------------------------------------------------------------
//визначення та виведення на екран поточного стану системи газопостачання
void __fastcall TForm_gas::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 15);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 16);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";

n=GetVariable(15);
if(n="0") Label_11->Caption="В нормі"; else Label_11->Caption="Виявлено витік
газу";
n=GetVariable(16);
if(n="0") Label_22->Caption="В нормі"; else Label_22->Caption="Виявлено дим";

}
```

КбПЗ_2025

Файл gas.h – бібліотека для файлу gas.cpp

```
#ifndef gasH
#define gasH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//---------------------------------------------------------------------------
class TForm_gas : public TForm
{
__published:      // IDE-managed Components
        TGroupBox *GroupBox1;
        TLabel *Label1;
        TLabel *Label_1;
        TLabel *Label13;
        TLabel *Label_11;
        TButton *Button1;
        TButton *Button2;
        TGroupBox *GroupBox2;
        TLabel *Label3;
        TLabel *Label_2;
        TLabel *Label5;
        TLabel *Label_22;
        TButton *Button3;
        TButton *Button4;
        TTimer *Status;
        void __fastcall Button2Click(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall Button4Click(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_gas(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm_gas *Form_gas;
//---------------------------------------------------------------------------
#endif
```

КОПІЯ 2025

Файл tv.cpp - керування дломашнім кінотеатром

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "tv.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_tv *Form_tv;
//---------------------------------------------------------------------------
__fastcall TForm_tv::TForm_tv(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm_tv::TrackBar1Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,32,TrackBar1->Position);
}
//---------------------------------------------------------------------------


void __fastcall TForm_tv::onClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,32,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_tv::offClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,32,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_tv::ComboBox1Change(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,33,23,ComboBox1->Text);
}
//---------------------------------------------------------------------------
//визначення та виведення стану телевізору

void __fastcall TForm_tv::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 33);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";

n=GetVariable(33);
ComboBox1->Text=n;

}
//---------------------------------------------------------------------------
```

КОПЗ-2025

Файл tv.h – бібліотека для файлу tv.cpp

```
#ifndef tvH
#define tvH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//---------------------------------------------------------------------------
class TForm_tv : public TForm
{
__published:      // IDE-managed Components
        TLabel *Label11;
        TLabel *Label_1;
        TLabel *torch;
        TTrackBar *TrackBar1;
        TButton *on;
        TButton *off;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label4;
        TComboBox *ComboBox1;
        TImage *Image4;
        TBevel *Bevel1;
        TLabel *Label3;
        TTimer *Status;
        void __fastcall TrackBar1Change(TObject *Sender);
        void __fastcall onClick(TObject *Sender);
        void __fastcall offClick(TObject *Sender);
        void __fastcall ComboBox1Change(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_tv(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm_tv *Form_tv;
//---------------------------------------------------------------------------
#endif
```

КОПІЯ 2025

Файл phone.cpp – керування телефонним зв'язком

```
#include <vcl.h>
#pragma hdrstop

#include "phone.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_phone *Form_phone;
//---------------------------------------------------------------------------
__fastcall TForm_phone::TForm_phone(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::TrackBar1Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,30,TrackBar1->Position);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::TrackBar2Change(TObject *Sender)
{
ClientSocket->SendLeviton_LPWAN(0,31,TrackBar2->Position);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::t_onClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,30,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::t_offClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,30,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::a_onClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,31,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::a_offClick(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,31,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_phone::Button3Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,30,23,Edit1->Text);
}
//---------------------------------------------------------------------------

//визначення та виведення стану телефонного зв'язку
void __fastcall TForm_phone::StatusTimer(TObject *Sender)
{
ShortString st;
int n;
```

КОПЗ_2025

```
ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 30);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 31);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";

}
//---------------------------------------------------------------------------
```

КбПЗ_2025

Файл phone.h – бібліотека для файлу phone.cpp

```
#ifndef phoneH
#define phoneH
//---------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//---------------------------------------------------------------------
class TForm_phone : public TForm
{
__published:      // IDE-managed Components
        TLabel *Label11;
        TLabel *Label17;
        TLabel *Label_1;
        TTrackBar *TrackBar1;
        TButton *t_on;
        TButton *t_off;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label4;
        TEdit *Edit1;
        TButton *a_on;
        TButton *a_off;
        TLabel *Label6;
        TTrackBar *TrackBar2;
        TLabel *Label7;
        TLabel *Label8;
        TLabel *Label9;
        TLabel *Label_2;
        TButton *Button3;
        TImage *Image4;
        TImage *Image1;
        TBevel *Bevel1;
        TLabel *Label3;
        TBevel *Bevel2;
        TLabel *Label5;
        TTimer *Status;
        void __fastcall TrackBar1Change(TObject *Sender);
        void __fastcall TrackBar2Change(TObject *Sender);
        void __fastcall t_onClick(TObject *Sender);
        void __fastcall t_offClick(TObject *Sender);
        void __fastcall a_onClick(TObject *Sender);
        void __fastcall a_offClick(TObject *Sender);
        void __fastcall Button3Click(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_phone(TComponent* Owner);
};
extern PACKAGE TForm_phone *Form_phone;
#endif
```

КОПЗ_2025

Файл security.cpp – керування системою безпеки

```
#include <vcl.h>
#pragma hdrstop

#include "security.h"
#include "CPU_LPWANSocket"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_security *Form_security;
//---------------------------------------------------------------------------
__fastcall TForm_security::TForm_security(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm_security::Button14Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,17,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button20Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,18,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button16Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,19,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button18Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,19,18,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button13Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,15,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button19Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,16,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button15Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,17,19,1);
}
//---------------------------------------------------------------------------

void __fastcall TForm_security::Button17Click(TObject *Sender)
{
ClientSocket->Send_LPWANCommand(0,18,19,1);
}
//---------------------------------------------------------------------------

//визначення та виведення стану системи безпеки
```

КбПЗ_2025

```
void __fastcall TForm_security::StatusTimer(TObject *Sender)
{
ShortString st;
int n;

ClientSocket->Send_LPWANStatusQuery();
st=_LPWANState->GetStateOnOff(0, 17);
if(st="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 18);
if(st="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";

st=_LPWANState->GetStateOnOff(0, 19);
if(st="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";
st=_LPWANState->GetStateOnOff(0, 20);
if(st="On") Label_4->Caption="Заблоковано"; else Label_4->Caption="Відкрито";


n=GetVariable(17);
if(n="0") Label_11->Caption="В нормі"; else Label_11->Caption="Спрацьовування";
n=GetVariable(18);
if(n="0") Label_22->Caption="В нормі"; else Label_22->Caption="Спрацьовування";
}
```

КбПЗ_2025

Файл security.h – бібліотека для файлу security.cpp

```
#ifndef securityH
#define securityH
//---------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Mask.hpp>
#include <ExtCtrls.hpp>
//---------------------------------------------------------------------
class TForm_security : public TForm
{
__published:      // IDE-managed Components
        TButton *Button12;
        TGroupBox *GroupBox1;
        TLabel *Label4;
        TLabel *Label_1;
        TLabel *Label13;
        TLabel *Label_11;
        TButton *Button13;
        TButton *Button14;
        TGroupBox *GroupBox2;
        TLabel *Label6;
        TLabel *Label_3;
        TButton *Button15;
        TButton *Button16;
        TGroupBox *GroupBox3;
        TLabel *Label8;
        TLabel *Label_4;
        TButton *Button17;
        TButton *Button18;
        TGroupBox *GroupBox4;
        TLabel *Label10;
        TLabel *Label_2;
        TLabel *Label12;
        TLabel *Label_22;
        TButton *Button19;
        TButton *Button20;
        TTimer *Status;
        void __fastcall Button14Click(TObject *Sender);
        void __fastcall Button20Click(TObject *Sender);
        void __fastcall Button16Click(TObject *Sender);
        void __fastcall Button18Click(TObject *Sender);
        void __fastcall Button13Click(TObject *Sender);
        void __fastcall Button19Click(TObject *Sender);
        void __fastcall Button15Click(TObject *Sender);
        void __fastcall Button17Click(TObject *Sender);
        void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public:     // User declarations
        __fastcall TForm_security(TComponent* Owner);
};
extern PACKAGE TForm_security *Form_security;
#endif
```

КОПЗ_2025

Файл about.cpp – довідка про програму

```
#include <vcl.h>
#pragma hdrstop

#include "about.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_about *Form_about;
//---------------------------------------------------------------------------
__fastcall TForm_about::TForm_about(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm_about::Button1Click(TObject *Sender)
{
Form_about->Close();
}
//---------------------------------------------------------------------------

void __fastcall TForm_about::FormCreate(TObject *Sender)
{
Memo1->Lines->Add("");
Memo1->Lines->Add("");
Memo1->Lines->Add("БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА");
Memo1->Lines->Add("");
Memo1->Lines->Add("на  тему:");
Memo1->Lines->Add("");
Memo1->Lines->Add("Програмне  забезпечення  IoT  інтелектуального  будинку  з
використанням мережі LPWAN");
Memo1->Lines->Add("");
Memo1->Lines->Add("");
Memo1->Lines->Add("");
Memo1->Lines->Add("Керівник: Петренюк В.І.");
Memo1->Lines->Add("");
Memo1->Lines->Add("Розробив: студент Удудяк Богдан Олексійович ");
Memo1->Lines->Add("                      гр. КІ-21-2");
Memo1->Lines->Add("");
Memo1->Lines->Add("");
Memo1->Lines->Add("м. Кропивницький 2025");
}
```

КОПЗ_2025

Файл about.h – бібліотека для файлу about.cpp

```
//---------------------------------------------------------------------------

#ifndef aboutH
#define aboutH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <jpeg.hpp>
//---------------------------------------------------------------------------
class TForm_about : public TForm
{
__published:      // IDE-managed Components
        TImage *Image1;
        TMemo *Memo1;
        TButton *Button1;
        void __fastcall Button1Click(TObject *Sender);
private:    // User declarations
public:           // User declarations
        __fastcall TForm_about(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TForm_about *Form_about;
//---------------------------------------------------------------------------
                                        #endif
```

КбПЗ_2025

Файл CPU_LPWANSocket.cpp – протокол X10

```
//------- -------------------

#include <basepch.h>

#pragma hdrstop

#include "CPU_LPWANSocket.h"

#pragma package(smart_init)
//------- -------------------
//

static inline void ValidCtrCheck(TCPU_LPWANSocket *)
{
    new TCPU_LPWANSocket(NULL);
}
//#include "OcelotStateUnit.cpp"
//------- -------------------
__fastcall TCPU_LPWANSocket::TCPU_LPWANSocket(TComponent* Owner)
    : TClientSocket(Owner)
{
_LPWANStateChangeNum=new T_LPWANStateChangeNum;
OcelotStateChangeNum=new TOcelotStateChangeNum;
_LPWANState=new T_LPWANState;
OcelotState=new TOcelotState;

//Початкова ініціалізація змінних
_LPWANStateChangeNum->Num=0;
OcelotStateChangeNum->Num=0;
FAuth=1;

// установлюємо номер порту
Port=63336;
SetVersion("");
OnConnect=MyOnConnect;
OnRead=MyOnRead;
OnDisconnect=MyOnDisconnect;
OnError=MyOnError;

}

__fastcall TCPU_LPWANSocket::~TCPU_LPWANSocket(void)
{
delete _LPWANStateChangeNum;
delete OcelotStateChangeNum;
delete _LPWANState;
delete OcelotState;

}
//------- -------------------
namespace CPU_LPWANSocket
{
    void __fastcall PACKAGE Register()
    {
         TComponentClass classes[1] = {__classid(TCPU_LPWANSocket)};
         RegisterComponents(" CPU-XA", classes, 0);
    }
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::MyOnConnect(System::TObject* Sender,
TCustomWinSocket* Socket)
{
char buf[16];
buf[0]='a';
memcpy(&buf[1],(Login).c_str(),5);
memcpy(&buf[6],(Password).c_str(),10);
SendBuf(buf,16);
```

КбПЗ_2025

```
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::MyOnDisconnect(System::TObject* Sender,
TCustomWinSocket* Socket)
{
SendBuf("d",1);
FAuth=2;
if (FOnChangeAuthStatus)
    FOnChangeAuthStatus(this,FAuth);
}

//------- -------------------
void __fastcall TCPU_LPWANSocket::MyOnError(System::TObject* Sender,
TCustomWinSocket* Socket, TErrorEvent ErrorEvent, int &ErrorCode)
{
if (FOnChangeAuthStatus)
    FOnChangeAuthStatus(this,5);
if (OnMyError)
    OnMyError(Sender,Socket,ErrorEvent,ErrorCode);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::MyOnRead(System::TObject* Sender,
TCustomWinSocket* Socket)
{

long size=Socket->ReceiveLength();
char *buff=(char *)malloc(size);
char *local_pointer; // локальний покажчик на місце в буфері з якого починається
команда
long Offset=0; // зсув про буферу, що прийшов
long CommandStrSize=0; // розмір що прийшов команди

Socket->ReceiveBuf(buff,size);
local_pointer=&buff[Offset];

if (size==0) goto end_label; // якщо нічого не прийшло, то про всякий випадок
чистимо буфер

NewLoop:

// Перевіряємо на те, що це дійсно команда, а не обривок якого-небудь логу
if ((( size-Offset)>5)&&(memcmp(local_pointer,"CPUXA",5)==0))
    {
    Offset=Offset+5;
    local_pointer=&buff[Offset];
    }
else
    {
    goto end_label;
    }

if (( size-Offset)<2)
    goto end_label;
else
    {
    memcpy(&CommandStrSize,local_pointer,2); // Довідалися довжину текстової
команди
    Offset=Offset+2;
    local_pointer=&buff[Offset];
    }
if (( size-Offset)<CommandStrSize) goto end_label; // Якщо шматок до кінця
залишився менше, ніж довжина команди, виходимо з функції


switch ( local_pointer[0] )
{
case  'a':
            if (CommandStrSize==2)
                {
```

КПЗ – 2025

```
                FAuth=local_pointer[1];
                if (FOnChangeAuthStatus)
                    FOnChangeAuthStatus(this,FAuth);
                };
            break;

case  'c': break;

case  'd':
            if (CommandStrSize==2)
                {
                FAuth=2;
                if (FOnChangeAuthStatus) FOnChangeAuthStatus(this,FAuth);
                };
            break;

case  'p':
                if (FOnReadOtherData)
FOnReadOtherData(this,&local_pointer[0],CommandStrSize);

case  'q':
            if (CommandStrSize==1034+3)
                {
                memcpy(&OcelotStateChangeNum->Num,&local_pointer[1],2);
                OcelotState->LoadInfoBuff(&local_pointer[3]);
                if (FOnReadOcelotStatusData)
                    FOnReadOcelotStatusData(this,true);
                }
            if (CommandStrSize==3)
                if (FOnReadOcelotStatusData)
                    FOnReadOcelotStatusData(this,false);
            break;

case  'x':
            if (CommandStrSize==259)
                {
                memcpy(&_LPWANStateChangeNum->Num,&local_pointer[1],2);
                _LPWANState->LoadInfoBuff(&local_pointer[3]);
                if (FOnRead10StatusData)
                    FOnRead10StatusData(this,true);
                }
            if (CommandStrSize==3)
                if (FOnRead10StatusData)
                    FOnRead10StatusData(this,false);
            break;


default : if (FOnReadOtherData)
FOnReadOtherData(this,local_pointer,CommandStrSize);
}
if (( size-Offset-CommandStrSize)>0) // якщо шматок блоку, що залишився, більше
0 (може в купі лежить ще одна програма)
    {
    Offset=Offset+CommandStrSize; // указуємо зрушення
    local_pointer=&buff[Offset];
    goto NewLoop; // Пішли на нове коло
    }

end_label:
free(buff);
}
//------- ------------------
bool __fastcall TCPU_LPWANSocket::SendBuf(char *buf,long size)
{
if (Active)
    {
    char *buff;
    long size_buff=size+7;
    buff=(char *)malloc(size_buff);
```

КПІ - 2025

```
    memcpy(buff,"CPUXA",5);
    memcpy(&buff[5],&size,2);
    memcpy(&buff[7],buf,size);
    Socket->SendBuf(buff,size_buff);
    free(buff);
    return true;
    }
else
    return false;
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::SendOcelotStatusQuery(void)
{
char buf[3];
buf[0]='q';
memcpy(&buf[1],&OcelotStateChangeNum->Num,2);
SendBuf(buf,3);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::Send10StatusQuery(void)
{
char buf[3];
buf[0]='x';
memcpy(&buf[1],&_LPWANStateChangeNum->Num,2);
SendBuf(buf,3);
}
//------- -------------------
unsigned short __fastcall TCPU_LPWANSocket::GetDataFromUnit(unsigned char
UnitNumber)
{
return OcelotState->GetDataFromUnit(UnitNumber);
}
//------- -------------------
unsigned char __fastcall TCPU_LPWANSocket::GetIO(unsigned char
UnitNumber,unsigned char Point)
{
return OcelotState->GetIO(UnitNumber,Point);
}
//------- -------------------
unsigned char __fastcall TCPU_LPWANSocket::GetUnitVer(unsigned char UnitNumber)
{
return OcelotState->GetUnitVer(UnitNumber);
}
//------- -------------------
unsigned char __fastcall TCPU_LPWANSocket::GetUnitType(unsigned char UnitNumber)
{
return OcelotState->GetUnitType(UnitNumber);
}
//------- -------------------
unsigned char __fastcall TCPU_LPWANSocket::GetUnitsCount(void)
{
return OcelotState->GetUnitsCount();
}
//------- -------------------
TDateTime __fastcall TCPU_LPWANSocket::GetCPUXADateTime()
{
return OcelotState->GetCPUXADateTime();
}
//------- -------------------
unsigned short __fastcall TCPU_LPWANSocket::GetVariable(unsigned char VarNumber)
{
return OcelotState->GetVariable(VarNumber);
}
//------- -------------------
unsigned short __fastcall TCPU_LPWANSocket::GetTimer(unsigned char TimerNumber)
{
return OcelotState->GetTimer(TimerNumber);
}
//------- -------------------
```

КОПИЯ 2025

```
TDateTime TCPU_LPWANSocket::GetTimerAsTime(unsigned char VarNumber)
{
  return UShortToTime(OcelotState->GetTimer(VarNumber));
}
//------- -------------------
TDateTime TCPU_LPWANSocket::GetVarAsTime(unsigned char VarNumber)
{
  return UShortToTime(OcelotState->GetVariable(VarNumber));
}
//------- -------------------
TDateTime TCPU_LPWANSocket::UShortToTime(unsigned short value)
{
  Word hour=div(value,100).quot;
  Word min=div(value,100).rem;
  try
  {
    return EncodeDateTime(1899, 12, 30, hour, min, 0, 0);
  }
  catch ( ... )
  { //12/30/1899 12:00 am
    return EncodeDateTime(1899, 12, 30, 12, 0, 0, 0);
  }
}
//------- -------------------
void TCPU_LPWANSocket::SetTimeAsVar(unsigned char VarNumber, TDateTime time)
{
  SetVariable(VarNumber,TimeToUShort(time));
}
//------- -------------------
void TCPU_LPWANSocket::SetTimeAsTimer(unsigned char VarNumber, TDateTime time)
{
  SetTimer(VarNumber,TimeToUShort(time));
}
//------- -------------------
unsigned short TCPU_LPWANSocket::TimeToUShort(TDateTime time)
{
   Word Hour, Min, Sec, MSec;
   DecodeTime(time, Hour, Min, Sec, MSec);
   return Hour*100+Min;
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::SetIO(unsigned char UnitNumber,unsigned char
Point,unsigned char Stat)
{
char buf[5];
buf[0]='c';
buf[1]=0;
buf[2]=UnitNumber;
buf[3]=Point;
buf[4]=Stat;
SendBuf(buf,5);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket:: SetDateTime(unsigned char day,unsigned char
mon,unsigned char year,unsigned char hour,unsigned char min)
{
char buf[7];
buf[0]='c';
buf[1]=1;
buf[2]=day;
buf[3]=mon;
buf[4]=year;
buf[5]=hour;
buf[6]=min;
SendBuf(buf,7);

}
//------- -------------------
```

КБГЭ_2025

```
void __fastcall TCPU_LPWANSocket::SetVariable(unsigned char VarNumber,unsigned
short Value)
{
char buf[5];
buf[0]='c';
buf[1]=2;
buf[2]=VarNumber;
memcpy(&buf[3],&Value,2);
SendBuf(buf,5);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::SetTimer(unsigned char TimerNumber,unsigned
short Value)
{
char buf[5];
buf[0]='c';
buf[1]=3;
buf[2]=TimerNumber;
memcpy(&buf[3],&Value,2);
SendBuf(buf,5);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::SendLeviton10(unsigned char house,unsigned
char key, unsigned char dim)
{
char buf[5];
buf[0]='c';
buf[1]=4;
buf[2]=house;
buf[3]=key;
buf[4]=dim;
SendBuf(buf,5);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::SendIr(unsigned short Value)
{
char buf[4];
buf[0]='c';
buf[1]=5;
memcpy(&buf[2],&Value,2);
SendBuf(buf,4);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::Send10Buff(unsigned char house,unsigned char
key, unsigned char repeat)
{
char buf[5];
buf[0]='c';
buf[1]=6;
buf[2]=house;
buf[3]=key;
buf[4]=repeat;
SendBuf(buf,5);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::Send10Command(unsigned char house,unsigned
char key, unsigned char CommandNum, unsigned char repeat)
{
char buf[6];
buf[0]='c';
buf[1]=7;
buf[2]=house;
buf[3]=key;
buf[4]=CommandNum;
buf[5]=repeat;
SendBuf(buf,6);
}
//------- -------------------
void __fastcall TCPU_LPWANSocket::GetInternalProtocolVersion(void)
```

КБПЗ_2025

```
{
char buf[1];
buf[0]='p';
SendBuf(buf,1);
}
//------- -------------------

void __fastcall T_LPWANStateChangeNum::Next(void)
{
if (Num==65534)
    Num=1;
else
    Num++;
};
//------- -------------------
__fastcall T_LPWANStateChangeNum::T_LPWANStateChangeNum(void)
{
Num=1;
};
//------- -------------------
AnsiString __fastcall T_LPWANState::GetStateOnOff(unsigned char house, unsigned
char key)
{
if (map[house][key]==true)
return "On";
else
return " --";
}
//------- -------------------
bool __fastcall T_LPWANState::GetOnOffStatus(unsigned char house, unsigned char
key)
{
if (map[house][key]==true)
return true;
else
return false;
}
//------- -------------------

bool __fastcall T_LPWANState::LoadInfoBuff(char *buff)
{
memcpy(map,buff,256);
return true;
}
//------- -------------------

void _fastcall TCPU_LPWANSocket::SetLogin(AnsiString str)
{
int len;
len=str.Length();
if (len<5)
    {
    for(int i=len;i<5;i++)
        str=str+"1";
    }
FLogin=str.SubString(1,5);
return;
}

void _fastcall TCPU_LPWANSocket::SetPassword(AnsiString str)
{
int len;
len=str.Length();
if (len<10)
    for(int i=len;i<10;i++)
        str=str+"1";
FPassword=str.SubString(1,10);
return;
}
```

КОПЗ_2025

Файл CPU_LPWANSocket.h - бібліотека для файлу CPU_LPWANSocket.cpp

```
//------- ------------------

#ifndef CPU_LPWANSocket
#define CPU_LPWANSocket
//------- ------------------
#include <SysUtils.hpp>
#include <Classes.hpp>
#include <ScktComp.hpp>

#include "OcelotStateUnit.h"


//------- ------------------
const int MajorVersion = 1;
const int MinorVersion = 1;
// Опис типів викликуваних подій
    typedef void __fastcall (__closure *TOnReadOcelotStatusData)(System::TObject
*Sender, bool HaveNewData);
    typedef void __fastcall (__closure *TOnRead10StatusData)(System::TObject
*Sender, bool HaveNewData);
    typedef void __fastcall (__closure *TOnReadOtherData)(System::TObject
*Sender, char *Data, long size);
    typedef void __fastcall (__closure *TOnChangeAuthStatus)(System::TObject
*Sender, char FAuth);
    typedef void __fastcall (__closure *TOnMyError)(System::TObject* Sender,
TCustomWinSocket* Socket, TErrorEvent ErrorEvent, int &ErrorCode);

class T_LPWANStateChangeNum {
public:           // User declarations
unsigned short Num;
void __fastcall Next(void);
__fastcall T_LPWANStateChangeNum(void);
};

class T_LPWANState {
private:
unsigned char ActiveKey[16];
public:           // User declarations
AnsiString __fastcall GetStateOnOff(unsigned char house, unsigned char key);
bool __fastcall GetOnOffStatus(unsigned char house, unsigned char key);
bool LightCommandMask[16][16];
bool UnitCommandMask[16][16];
bool map[16][16];
bool __fastcall LoadInfoBuff(char *buff);
};


class PACKAGE TCPU_LPWANSocket : public TClientSocket
{
private:
    AnsiString FVersion;
    AnsiString FLogin;
    AnsiString FPassword;
    unsigned char FAuth;
    TOcelotStateChangeNum *OcelotStateChangeNum;
    T_LPWANStateChangeNum *_LPWANStateChangeNum;
    TOcelotState *OcelotState;

    // перевірка правильності уведення даних
    void __fastcall SetLogin(AnsiString str);
    void __fastcall SetPassword(AnsiString str);
    void __fastcall SetVersion(AnsiString)
        {
        FVersion=(AnsiString)MajorVersion+"."+(AnsiString)MinorVersion;
        }
```

КПЗ_2025

```
    void __fastcall MyOnError(System::TObject* Sender, TCustomWinSocket* Socket,
TErrorEvent ErrorEvent, int &ErrorCode);
    void __fastcall MyOnConnect(System::TObject* Sender, TCustomWinSocket*
Socket);
    void __fastcall MyOnRead(System::TObject* Sender, TCustomWinSocket* Socket);
    void __fastcall MyOnDisconnect(System::TObject* Sender, TCustomWinSocket*
Socket);
    bool __fastcall SendBuf(char *buf,long size);
    // Покажчики на мої власні події
    TOnReadOcelotStatusData FOnReadOcelotStatusData;
    TOnRead10StatusData FOnRead10StatusData;
    TOnReadOtherData FOnReadOtherData;
    TOnChangeAuthStatus FOnChangeAuthStatus;
    TOnMyError FOnMyError;

    TDateTime UShortToTime(unsigned short value);
    unsigned short TimeToUShort(TDateTime time);
protected:

public:
    // конструктор і деструктор класу
    __fastcall TCPU_LPWANSocket(TComponent* Owner);
    __fastcall ~TCPU_LPWANSocket(void);
    T_LPWANState *_LPWANState;
    // запит на одержання даних про статуси
    void __fastcall SendOcelotStatusQuery(void);
    void __fastcall Send10StatusQuery(void);
    // Перенос методов з модуля
    unsigned short __fastcall GetDataFromUnit(unsigned char UnitNumber);
    unsigned char __fastcall GetIO(unsigned char UnitNumber,unsigned char
Point);
    unsigned char __fastcall GetUnitVer(unsigned char UnitNumber);
    unsigned char __fastcall GetUnitType(unsigned char UnitNumber);
    unsigned char __fastcall GetUnitsCount(void);
    TDateTime __fastcall GetCPUXADateTime();
    unsigned short __fastcall GetVariable(unsigned char VarNumber);
    TDateTime GetVarAsTime(unsigned char VarNumber);
    TDateTime GetTimerAsTime(unsigned char VarNumber);
    void SetTimeAsVar(unsigned char VarNumber, TDateTime time);
    void SetTimeAsTimer(unsigned char VarNumber, TDateTime time);
    unsigned short __fastcall GetTimer(unsigned char TimerNumber);
    // Sending commands
    void __fastcall SetIO(unsigned char UnitNumber,unsigned char Point,unsigned
char Stat);
    void __fastcall SetDateTime(unsigned char day,unsigned char mon,unsigned
char year,unsigned char hour,unsigned char min);
    void __fastcall SetVariable(unsigned char VarNumber,unsigned short Value);
    void __fastcall SetTimer(unsigned char TimerNumber,unsigned short Value);
    void __fastcall SendLeviton10(unsigned char house,unsigned char key,
unsigned char dim);
    void __fastcall SendIr(unsigned short Value);
    void __fastcall Send10Buff(unsigned char house,unsigned char key, unsigned
char repeat);
    void __fastcall Send10Command(unsigned char house,unsigned char key,
unsigned char CommandNum, unsigned char repeat);
    // Функції додаткових даних, що відносяться до програми
    void __fastcall GetInternalProtocolVersion(void);
__published:
    __property unsigned char AuthStatus = {read=FAuth};
    __property AnsiString Login = {read=FLogin, write = SetLogin};
    __property AnsiString Password = {read=FPassword, write = SetPassword};
    __property AnsiString Version = {read=FVersion, write = SetVersion};
    // мої власні події

    __property TOnReadOcelotStatusData OnReadOcelotStatusData = {read =
FOnReadOcelotStatusData, write = FOnReadOcelotStatusData};
    __property TOnRead10StatusData OnRead10StatusData = {read =
FOnRead10StatusData, write = FOnRead10StatusData};
```

КОПІЯ 2025

```
    __property TOnReadOtherData OnReadOtherData = {read = FOnReadOtherData,
write = FOnReadOtherData};
    __property TOnChangeAuthStatus OnChangeAuthStatus = {read =
FOnChangeAuthStatus, write = FOnChangeAuthStatus};
    __property TOnMyError OnMyError = {read = FOnMyError, write = FOnMyError};

};
//------- -------------------

#endif
```

КбПЗ_2025