

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
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи
інтелектуального дому з використанням технології LCN”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-1
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Бутаков В.О.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2023 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Бутакову Владиславу Олександровичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Смірнова Тетяна Віталіївна, канд. техн. наук</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 32-13 від 04.08.2023 року | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <i>10.12.2023 р.</i> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <i>Метою розробки є дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN</i> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Економічна ефективність розробленої програми.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | <i>5. Впровадження системи в промислову експлуатацію</i> | |
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | | | | | | | | | | |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | | | | | | | | | | |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | | | | | | | | | | |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | | | | | | | | | | |
| <i>5. Впровадження системи в промислову експлуатацію</i> | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <i>Наукова новизна</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> | | | | | | | | | | |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> | | | | | | | | | | |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Бутаков В.О. Дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального дому з використанням технології LCN.

Метою розробки є дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN.

Об'єктом дослідження є процес інтелектуального дому з використанням технології LCN.

Предметом дослідження є методи інтелектуального дому з використанням технології LCN.

Методи дослідження базуються на методах Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуального дому з використанням технології LCN.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Builder C++.

Ключові слова: комп'ютерні науки, інтелектуальний дім, LCN

ABSTRACT

Butakov V.O. Research and software implementation of the intelligent home system using LCN technology. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the intelligent home system using LCN technology.

The purpose of the development is the research and software implementation of the intelligent home system using LCN technology.

The object of research is the process of an intelligent home using LCN technology.

The subject of research is the methods of the intelligent home using LCN technology.

Research methods are based on Internet of Things methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the intelligent home system using LCN technology.

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 tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

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

Keywords: computer science, intellectual house, LCN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	35
3.3 Розробка функціональної схеми	41
3.4 Розробка діаграми процесів.....	44
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	62
6 НАУКОВА НОВИЗНА	70

						ВКРМ-122.23.0003.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата		Літ.	Аркуш	Аркушів
Розроб.	Бутаков В.О.				Дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN	М	1	110
Перев.	Смірнова Т.В.					ЦНТУ КН-22М-1		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	71
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	71
7.2 Розрахунок трудомісткості розробки програмної продукції.....	73
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	75
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	79
7.5 Визначення собівартості розробки та ціни програмної продукції.....	84
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	87
7.7 Визначення експлуатаційних витрат.....	87
7.8 Визначення економічної ефективності програмної продукції.....	89
7.9 Висновок.....	91
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	92
8.1 Вступ.....	92
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	94
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	95
8.4 Розробка заходів з умов поліпшення охорони праці.....	98
8.5 Розрахунок занулення.....	99
9 ОСНОВНІ ВИСНОВКИ.....	102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	104

ВСТУП

Актуальність теми. Поняття «Розумний будинок» або «Інтелектуальний дім» стає сьогодні досить звичним не тільки в колі технічних фахівців, але й серед людей, далеких від техніки автоматизації будинків. У поняття «будинок» ми можемо включити практично будь-який будинок: житлове приміщення, офіс, виробничий корпус. Всі ці спорудження поєднує наявність великого числа самих складних інженерних систем і комунікацій, які перебувають у постійній взаємодії з людиною й навколишнім середовищем. І ми, перебуваючи в кожному із цих будинків або приміщень, очікуємо наявності комфортних для нашого життя й роботи умов, незалежно від погоди на вулиці, температури, вологості, освітленості й т.д. Яким повинен бути гарний сучасний будинок? Насамперед, зручним. Система «Розумний будинок» надає поняттю комфорту новий, не досяжний раніше зміст, суть якого в тому, що будинок стає «слухняним», він кориться волі свого хазяїна й навіть угадує його бажання, «підкується» про нього.

Будь-який сучасний будинок оснащений технічними пристроями, які допомагають людині в його роботі й побуті, забезпечує затишок і безпеку, проведення дозвілля або максимальний комфорт у роботі. У той же час, кожна з використовуваних для цієї мети систем окремо складна в керуванні. Навіть пульт від телевізора має кілька десятків кнопок. Тому людина не може повністю ефективно контролювати роботу технічних систем будинку, використовувати кілька пристроїв одночасно, створюючи для себе підходяще середовище перебування. Багато хто погодяться із твердженням, що достаток приладів у будинку нерідко не спрощує, а ускладнює життя його мешканцям. Саме для об'єднання всіх систем керування технікою, що забезпечує умови перебування людини в будинку й адаптації їх для конкретного користувача, і була створена система «розумний будинок».

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN.

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

– Огляд існуючих систем інтелектуального дому з використанням технології LCN.

– Дослідження системи інтелектуального дому з використанням технології LCN.

– Програмна реалізація системи інтелектуального дому з використанням технології LCN.

Об'єктом дослідження є процес інтелектуального дому з використанням технології LCN.

Предметом дослідження є методи інтелектуального дому з використанням технології LCN.

Методи дослідження базуються на методах Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуального дому з використанням технології LCN.

– Розроблено вітчизняний продукт інтелектуального дому з використанням технології LCN, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального дому з використанням технології LCN.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГІЗ-2023

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для управління інтелектуальним домом. Компоненти комплексу «Інтелектуальний дім» розглянемо більш конкретно й докладно на прикладі системи LCN (Local Control Network).

Сьогодні програмне забезпечення LCN повністю сумісне з операційною системою WINDOWS. І не становить ніякої праці швидко вивчити правила її практичного застосування для реалізації конкретних проектів. Надзвичайно висока надійність і, одночасно, простота практичного застосування сприяли активному просуванню LCN у країнах Європи, Америки, Азії. А в самій Німеччині LCN займає приблизно 40% ринку систем автоматизації будинків.

Система LCN є децентралізованою, і її чудовою особливістю є можливість розвитку й розширення мережі зі збільшенням кількості реалізованих функцій. Споживач може спочатку придбати всього один-два модуля для реалізації якихось початкових функцій керування, а потім, здобуваючи додаткові модулі, нарощувати можливості системи у відповідності зі своїми запитами.

Шинні модулі є головним елементом LCN-шини. Кожний модуль має мікропроцесор, що обмінюється інформацією з іншими модулями, органами керування й пристроями візуалізації, об'єднаними в єдину мережу.

1.2 Область застосування

Областю застосування системи, яка розробляється у результаті виконання магістерського проектування є «інтелектуальний дім» – це комплексна система, що здатна об'єднати в єдину інтелектуальну мережу всі побутові прилади й

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

інженерні системи будинку, відповідає за їхню погоджену роботу й ощадливе використання всіх видів ресурсів, і, насамперед, – енергетичних.

Перелічимо основні функції, які пропонує споживачеві система «Інтелектуальний дім».

1. Керування жалюзі, рольставнями, автоматичними воротами. Система "Інтелектуальний дім" може автоматично управляти всіма рольставнями, жалюзі й автоматичними воротами, залежно від закладеного сценарію. Наприклад: при під'їзді до будинку автоматичні ворота – відкриваються. Або: з настанням вечора рольставні або жалюзі на вікнах самі закриваються, а при настанні ранку – відкриваються.

2. Керування пристроями безперебійного енергозабезпечення. Відключення електроенергії нерідко приведе до небажаних наслідків і аварій. Система "Інтелектуальний дім" автоматично дозволить перемкнути частину обраних споживачем електропристоїв на резервне живлення.

3. Керування освітленням. Створення світлових і колірних сцен (сценаріїв). За допомогою функції керування світлом можливо змінювати освітленість, розподіл світлових потоків і колірну гаму приміщення, створювати затишок і романтичний настрій. Система "Інтелектуальний дім" дозволяє включати керування світлом у різні сценарії, що дозволить створити ту атмосферу, що захочете, одним натисканням клавіші вимикача.

4. Керування кліматом у приміщенні. Система "Клімат-контролю" дозволить управляти всіма кліматичними пристроями (казан, кондиціонер, вентиляція й т.д.) з будь-якої крапки в будинку із сенсорних панелей, пристроїв регулювання клімату, віддалено – через інтернет, мобільний телефон.

5. Контроль виникнення надзвичайних ситуацій (протікання води, загоряння й т.п.). У випадку виявлення у Вас будинку надзвичайної ситуації, про яку просигналізує датчик, система "Інтелектуальний дім" автоматично перекриє воду й сповістить про те, що трапилося.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

6. Керування опаленням, у тому числі в аварійних ситуаціях. Якщо під час відсутності в будинку протекли труби водопостачання, "Інтелектуальний дім" визначить, що відбулася протікання й перекриє подачу води. А після подзвонить на мобільний телефон і повідомить про аварію. Якщо протікання відбулася поруч із електроустановками, то вони автоматично будуть знеструмлені.

7. Мультирум – керування розподілом звуку в приміщеннях. Мультирум – це система розподілу звуку від декількох джерел аудіо– і відеозображення, що перебувають в одному місці; причому тип джерела й рівень гучності вибирається індивідуально з кожної кімнати.

8. Охорона, відеоспостереження з керованим режимом запису відеоінформації. Запис зображення з відеокамер виробляється, якщо під час відсутності відбулося несанкціоноване вторгнення. Згодом можете переглянути зображення з кожної з них. Система "Інтелектуальний дім" дозволяє включати запис тільки в тому випадку, якщо були задіяні датчики руху, і немає необхідності вести постійний відеоархів.

9. Керування доступом у будинок і його окремі приміщення. Можливі установки в систему охоронної сигналізації таких систем доступу, як зчитування сітківки ока, розпізнавання голосу, зняття відбитків пальців і т.д.

10. Обслуговування прилягаючих до будинку територій. Крім охорони прилягаючої до будинку території, «Інтелектуальний дім» дозволяє управляти автоматичним обігрівом прилягаючих до будинку доріжок, країв дахів, запобігаючи їх зледенінню.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У даному розділі розглянемо основні технології на які будуються інтелектуальні будинки.

LanDrive

LanDrive – це універсальна платформа для побудови класичних шинних розподілених систем управління в системах «розумний будинок», в автоматизації будинків. LanDrive – найбільш доступна на сьогоднішній день платформа для побудови шинних розподілених систем керування внутрішнім і вуличним освітленням, силовими навантаженнями, електроприладами, а також такими системами, як опалення, кондиціонування, вентиляція, охоронна сигналізація, контроль доступу й утікань води. Також можливе керування аудіо– і відеотехнікою, домашніми кінотеатрами, жалюзі, рольставнями, шторами, воротами, насосами, двигунами. В основному орієнтована на застосування в складі «розумного будинку», але останнім часом всі частіше застосовується в системах обліку й заощадження енергоресурсів, контролю доступу, охоронно-пожежних системах.

Структурно система складається із центрального контролера й виконавчих модулів, зв'язаних між собою польовою шиною (мережею). До виконавчих модулів підключаються кероване устаткування.

Для взаємодії система використовує на фізичному рівні стандарт RS-485. Для взаємодії на прикладному рівні використовується широко відомий протокол Modbus/RTU. Устаткування підтримує швидкість обміну інформацією в режимах 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 біт/с, що, досить для

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

миттєвого виконання команд. Максимальна кількість пристроїв, керованих одним головним контролером (ПЛК) – 100. Можливе об'єднання декількох головних контролерів в одну мережу. Через використання стандартних протоколів можливе включення в систему устаткування сторонніх виробників, також можливе використання пристроїв серії LanDrive в інших системах.

Елементи системи:

– Центральний керуючий контролер, що дозволяє управляти всіма пристроями по заданих сценаріях.

– Релейні модулі, що забезпечують комутацію потужного навантаження до 1,2 кВт, а також опитування датчика типу «сухий контакт».

– Дімміруючі модулі, що забезпечують можливість плавної зміни потужності в навантаженні.

– Модулі інфрачервоного зв'язку, що забезпечують прийом команд із пультів дистанційного керування й передачу збережених команд на побутові пристрої.

– Модулі з 4-ма цифровими входами, що забезпечують опитування декількох датчиків.

– Модулі розширення з аналоговими входами-виходами, що забезпечують опитування аналогових датчиків, наприклад, температури, і керування аналоговими пристроями.

– Модулі й датчики збору інформації: температури, вологості, руху.

– Опціональні комунікаційні модулі GSM.

– Блоки живлення всієї системи й центрального контролера.

У цій системі необхідне програмування тільки головного контролера. Для програмування використовується спеціалізоване програмне забезпечення – LanDrive Configurator Pro.

Система може управлятися за допомогою великого переліку програмного забезпечення:

– SCADA-додатки.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– OPC-сервери Modbus.

– Спеціальні додатки для керування будинками, розумними будинками, такі як IridiumMobile, IdomMedia.















Весь перелік додатків дозволяє здійснювати оперативне візуалізоване керування розумним будинком, у тому числі:

- керування системою LanDrive із сенсорних моніторів, панелей;
- керування системою LanDrive з кишенькових комп'ютерів;
- керування системою LanDrive з I-Phone, I-Pod, I-Pad;
- керування системою LanDrive через Інтернет по захищеному каналі;
- об'єднання безлічі центральних контролерів SPIDER у єдину мережу;
- створення власних віртуальних панелей керування будинком;
- можливість інтеграції з устаткуванням сторонніх виробників.

iRidium

iRidium Mobile – програмний комплекс для керування системами розумний будинок з мобільних пристроїв iPhone, iPad, iPod touch і будь-яких пристроїв під керуванням Windows XP/Vista/7, Windows Mobile/Сe.

Таблиця 2.1 – Устаткування яким може управляти iRidium

	AMX		SHTURMAN		AV CONTROL			SMART-BUS
	EIB/KNX		DOMINTELL		X10		Beta	HELVAR
	MODBUS		INSYTE		INELS		Beta	LUTRON
					TECO		Beta	CLIPSAL

С допомогою iRidium AV CONTROL Можна управляти практично будь-яким AV устаткуванням (TV, DVD, Bluray, проектори, AV ресивери, медіа-сервери, відеореєстратори, відеокамери, пристрої типу Global Cache, iRTrans і багато хто інші).



Рисунок 2.1 – Інтерфейс користувача iRidium на iPhone



Рисунок 2.2 – Інтерфейс користувача iRidium на iPad

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.23.0003.00.00.ПЗ

Арк.

13

Програма iRidium AV CONTROL необхідна, якщо потрібно реалізувати керування домашнім кінотеатром, мультирумом або просто управляти AV апаратурою у всьому будинку. Можливості програми iRidium AV CONTROL також включені в усі інші програми (для KNX, ModBus, Domintell, SmartBus і т.д.).

Склад програмного комплексу iRidium:

- iRidium Client.
- iRidium Client.
- iRidium Gate.
- iRidium GUI.
- iRidium Transfer.
- iRidium Wizard.

iRidium Client – головна частина програмного комплексу, запускає на iPhone/iPad, Windows пристрої підготовлений графічний інтерфейс. Відповідає за взаємодію запущеного на пристрої інтерфейсу з контролером і устаткуванням.

iRidium GUI Editor – графічний редактор, що дозволяє створювати будь-які графічні інтерфейси для панелі керування. Містить бібліотеку графічних елементів і типові інтерфейси для більше швидкого налаштування. У графічному редакторі відбувається прив'язка каналів і команд до керуючих кнопок, слайдерів і інших елементів.

iRidium Wizard – перетворить інтерфейс, створений в iRidium GUI Editor у формат *.igr для його наступного завантаження на iPhone/iPad. Підготує файл запуску для Windows додатка. Також виконує перетворення файлів з інших форматів AMX TPDesign4, Pronto Edit Professional.

iRidium Transfer – необхідний тільки для iPhone/iPad. Дозволяє завантажити готовий інтерфейс на дані пристрої. А також перевіряє наявність ліцензії для iPhone/iPad.

iRidium Gate – необхідний для взаємодії між iPhone, iPad, Windows панелями й деякими типами устаткування (KNX, Domintell, X10, SmartBus).

						ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			14

Наприклад, для керування устаткуванням KNX використовується IP Router типу JUNG IPR 100 REG. При такому способі з'єднання існує два обмеження:

1. Тільки одне TCP з'єднання. У результаті чого при додаванні другого, третього iPhone/iPad необхідно додавати додаткові IP Router. Це приводить до сильного подорожчання.

2. При некоректному розриві з'єднання IP Router у плинні декількох хвилин не реагує на команди керування. Що зовсім не влаштовує більшість клієнтів, тому що вони сприймають це за недоробку й помилку.

iRidium Gate усуває дані проблеми. iRidium Gate встановлюється на будь-який комп'ютер і забезпечує множинне з'єднання через один IP Router, а також відповідає за постійне з'єднання. Що робить керування з iPhone, iPad стабільним, надійним і істотно знижує вартість при додаванні другого, третього й більше iPhone/iPad пристроїв.

Переваги iRidium:

– Нативний код. Програма iRidium написана мовою C++ і Objective C. Це гарантує швидкість і стабільність роботи додатка. Для роботи програми не потрібні додаткові сервера й додаткові програмні модулі для контролерів. Взаємодія з контролером і устаткуванням відбувається з iPhone/iPad прямо через WiFi і Інтернет.

– iRidium OpenGL. Спеціально для роботи із графікою розроблено iRidium OpenGL – графічне ядро, на основі технології OpenGL, що забезпечує швидку роботу інтерфейсів, а також безліч спецефектів для кнопок, сторінок, тексту.

– Чудові графічні інтерфейси. iRidium GUI дозволяє створити дуже гарні й динамічні інтерфейси. При проектуванні свого будинку, яхти, квартири люди прагнуть створити неповторний дизайн, особливу атмосферу. iRidium дозволяє додати цю індивідуальність у панелі керування. Можна створити дуже більші інтерфейси з безліччю анімаційних ефектів. iRidium гарантує їх швидку й стабільну роботу на будь-якому iPhone, iPad.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Завантаження інтерфейсів з інших систем. Можна використовувати вже готові інтерфейси, створені в редакторах AMX TP4 і Pronto Edit Professional . Це полегшить роботу. Використовуючи iRidium Wizard, легко завантажити готові інтерфейси, створені в інших редакторах.

– Завантаження команд із ETS проекту для KNX. iRidium GUI містить ДУЖЕ корисну функцію – завантаження команд із проекту ETS для KNX, а також з подібних файлів для Domintell, Insyte і SmartBus. залишиться тільки методом drag&drop перетягнути завантажені канали й команди на керуючі кнопки, слайдери, списки й насолоджуватися керуванням з iPhone/iPad.

– Бібліотека графічних елементів і безкоштовні типові дизайни. iRidium GUI містить у собі бібліотеку графічних елементів. А також підготовлено кілька безкоштовних типових дизайнів. Це дозволяє дуже швидко настроїти керування розумним будинком, домашнім кінотеатром за допомогою iRidium.

– Підтримка медіа-сервера XBMC. При покупці iRidium для будь-якого устаткування одержуєте можливість управляти медіа-сервером XBMC. У найближчих версіях iRidium буде включена підтримка інших популярних медіа-серверів – Apple TV, Windows MC, Voxxe.

– Один інтерфейс для будь-якої платформи. Інтерфейс, створений в iRidium GUI, буде працювати на будь-якому пристрої (iPhone, iPad, Windows панелі) однаково й без доробок. Також працює для Linux і Android.

– Просте настроювання для устаткування AMX. Немає нічого простіше, ніж перетворити iPhone/iPad у панель керування AMX системою. Тільки 3 кроки. Установити iRidium з App Store. Конвертувати готовий проект із TP4 за допомогою iRidium Wizard. Завантажити проект на iPhone/ iPad за допомогою iRidium Transfer. Якщо проект не підготовлений спеціально під дозвіл iPhone/iPad наш iRidium Wizard сам піджене проект під потрібний розмір.

– Підтримка інерційних списків. Використовуючи інерційні списки, дизайн стає зручним для користувача. Дана можливість реалізована нами навіть

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

для системи АМХ. При цьому настроювання інерційних списків відбувається безпосередньо в рідному редакторі від АМХ.

– iRidium AV Control + Global Cache + iPad – краще рішення для керування домашнім кінотеатром і АV устаткуванням. Рішення iRidium AV Control + Global Cache + iPad повністю заміняє інші аналогічні рішення на базі АМХ, Philips Pronto і інших систем. Вартість такого рішення в 2 і більше раз нижче, а графічні можливості навіть кращі. За рахунок тісної інтеграції з устаткуванням Global Cache процес настроювання кінцевого проекту дуже простий і не займає багато часу.

– Безкоштовна ліцензія для інсталяторів. Будь-яка компанія-інсталятор може одержати по одній безкоштовній версії для iPhone/iPad і Windows пристрою.

Інші технології

Крім перерахованих вище продуктів, існують ще й наступні технології:

– LCN – німецька система автоматизації як для домашньої так і для промислової автоматизації. Повністю розподілений інтелект. Як середовище передачі використовується звичайне електричне проведення перетином жив 1,5 або 2,5 мм². Реалізується керування практично будь-яким обладнанням. Оптимальне співвідношення ціна/можливості.

– ВРТ – система домашньої автоматизації з розподіленим інтелектом, що використовує закритий протокол передачі даних. Як шина використовується стандартна кручена пара UTP. Реалізується керування освітленням, домашньою автоматикою, кондиціонуванням, опаленням, інженерною й охоронною сигналізація, домофонією. Основні мотиви вибору даної системи – низька вартість устаткування, простота монтажу й пуско-налаштування. Система сумісна з будь-якими типами електровстановочних виробів.

– MyHome SCS Інтегрується за допомогою OpenWebNet-шлюзу із системами від різних виробників.

– EIB (European Installation Bus – «Європейська інсталяційна шина»).

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

5-категорії (Cat.5), довжина якого в одному сегменті може становити 1000 м. Таких сегментів в одну систему можна поєднувати до 255. У мережі використовується 36В змінного струму. Протокол C-Bus використовується в Австралії, Новій Зеландії, Азії, на Близькому Сході, у Росії, США, Південній Африці, Великобританії й іншим частинам Європи, включаючи Грецію, Прибалтику, Румунію й інші країни. У США C-Bus представлений під маркою «Square Clipsal». Протокол C-Bus був створений Clipsal Integrated Systems для використання в системах домашньої автоматизації й системах керування освітленням будинків.

– Helvar – для систем керування освітленням використовує протокол DALI і DSI.

– AMX – система домашньої автоматизації однойменної компанії. Централізована. Протоколи закриті. Споконвічно застосовувалися власні шини передачі даних. Нові лінійки устаткування AMX використовують для передачі стандартні протоколи Ethernet, Wi-Fi і Zigbee. Має шлюзи для сполучення з іншими системами (EIB, LON і ін.).

– Crestron – протокол керування систем автоматизації й мультимедійних систем. Розроблений компанією Crestron (США). Протокол закритий. Основний конкурент – AMX (США).

– X10 – протокол керування електроприладами. Сигнал передається по електричним проводам або в радіодіапазоні. Недоліки – низька швидкість передачі інформації й перешкодозахищеність, проблема помилкового спрацьовування, відсутність зворотного зв'язка приймача з передавачем, можливі конфлікти пристроїв X10 різних виробників і несанкціонований доступ до пристроїв X10 по електромережі.

– Z-wave запатентований бездротовий протокол зв'язку, розроблений для домашньої автоматизації, зокрема для контролю й керування на житлових і комерційних об'єктах. Технологія використовує малопотужні й мініатюрні радіочастотні модулі, які вбудовуються в побутову електроніку й різні пристрої,

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

такі як освітлення, опалення, контроль доступу, розважальні системи й побутову техніку.

– LUXOR – система локального керування освітленням і кліматом. Використовується звичайний електричний кабель і вимикачі. Система виробляється Theben AG. Протокол закритий. Основні мотиви вибору даної системи – низька вартість устаткування, простота монтажу й пусконалаштування. Система сумісна з будь-якими типами електроустановочних виробів.

– Smart Bus – шина, розроблена в Канаді. Дана система розумний будинок є ідеальною в сполученні відносини ціна-якість.

– ONE-NET – відкритий протокол бездротової мережі передачі даних, розроблений для цілей автоматизації будинків і керування розподіленими об'єктами.

– R-BUS – шина розроблена спільно з відкритим протоколом передачі даних, призначена для автоматизації великих і малих споруджень.

– DOMINTELL – централізована система домашньої, офісної й готельної автоматизації. Використовує інтерфейс RS485 для обміну даними між модулями. Є шлюзи Ethernet TCP, UDP, RS232, B&O, DMX(керування світлом). Відкрита система команд керування "Light Protocol". Була розроблена в 1999 році в Бельгії.

– 1-Wire – технологія, що дозволяє зв'язати багато датчиків і прилади в одну мережу, керування в якій на себе бере персональний комп'ютер. Для передачі даних у такій мережі використовується всього одне проведення. Відрізняється дешевиною й простотою установки.

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

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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 управляемого компонента, 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 застосунку 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 випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуального дому з використанням технології LCN.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) 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 методика 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 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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Для опису функціонування системи надамо повний опис технології LCN.

LCN – німецька система автоматизації як для домашньої так і для промислової автоматизації. Повністю розподілений інтелект. Як середовище передачі використовується звичайне електричне проведення перетином жил 1,5 або 2,5 мм². Реалізується керування практично будь-яким устаткуванням. Оптимальне співвідношення ціна/можливості.

Шина LCN (Local Control Network) є шиною мультимастером і функціонує за принципом шин EIB і LON, так що кожний компонент може самостійно брати участь в обміні даними. Центральний пристрій при цьому не потрібно. Однак між LCN і іншими шинними системами існує концептуальна відмінність.

Згідно даним виробника, система LCN задовольняє потреби клієнтів при будівництві як приватних будинків, так і великих промислових об'єктів.

З 1994 року понад 1000 об'єктів були оснащені системою LCN (житлові комплекси, об'єкти середнього й великого бізнесу).

Як лінія передачі система LSN використовує додаткову жилу стандартного кабелю марки NYM. Це дозволяє заощаджувати на прокладці додаткового шинного кабелю й не вимагає проведення додаткової мережі. Передача даних відбувається в первинній смузі частот.

Принцип передачі даних

Для реалізації процесу обміну даними необхідна наявність середовища передачі. Дотепер розглядалися такі лінії передачі, як кручена пара, радіочастоти або модульований сигнал несучої частоти, переданий по електромережі напругою 220 В. У системі LCN дані передаються по додатковій жилі стандартного проведення NYM. Тобто при монтажі системи використовується пятижильний

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

кабель NYM. Якщо він є в наявності, виходить, будинок підготовлений до установки сучасного шинного встаткування. Передача даних відбувається в первинній смузі частот. При цьому відмовляються від використання несучої частоти, а швидкість передачі даних становить 9600 бод. При такій швидкості передачі система LCN може відправляти до 100 телеграм у секунду. Радіус дії системи становить 1000 метрів. Цього цілком достатньо для обслуговування великого житлового будинку. Максимальне число LCN модулів, які можуть бути з'єднані один з одним таким способом, становить 254.

Уже із цього можна зробити вивід, що система LCN підходить для застосування не тільки в житловому будинку, але й у великих об'єктах цільового призначення з більшим числом поверхів і кількістю приміщень до 100.

Кожний модуль LCN черпає енергію із власного блока живлення. Це значить, що модулі є взаємно незалежними. Таким чином, при бажанні можна приступитися до установки шини з одним-єдиним модулем. Існує можливість за допомогою високої щільності інтеграції помістити сенсорику, акторику й блок живлення в один модуль.

Практична інсталяція

Така висока інтеграція дозволяє фахівцеві проводити інсталяцію тим способом, до якого він звик. У тому місці, де раніше встановлювався вимикач, тепер розташовується модуль LCN.

Однак при будівництві необхідно передбачити глибину розеток для убудованого монтажу, щоб вони були сумісні з модулем діаметром 50 мм і глибиною 20 мм.

Старі види вимикачів замінюються клавшами. Якщо зі зворотної сторони стіни в прихожій розташована клавша, вона також приєднується до убудованого в кімнаті модулю.

Живильні проведення світильників (максимум дволампових) підключаються безпосередньо до двох 230-вольтових виходів LCN-модуля.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Таким чином, без зміни монтажу системи клієнт одержує ряд нових функцій:

- Плавне регулювання світла із двох незалежних виходів з будь-яким часом діммування.
- Наявність входу для ІЧ-дистанційного керування з $16 \times 3 = 48$ функціями.
- Динамічне групове формування.
- Рахункові й арифметичні функції керування.
- Система візуалізації (вивід інформації на табло).
- Миготлива лампа як датчик сигналу для людей з ушкодженням слуху, наприклад замість дверного дзвінка.
- Облік і обробка даних.

У приватному будинку напевно затребувані не всі можливості, які пропонує система LCN. Проте, фахівець із монтажу увесь час одержує замовлення від клієнтів на нестандартне виконання роботи. Це веде до збільшення зобов'язань перед клієнтом і, крім того, демонструє професіоналізм і компетенцію фахівця.

Технологія LCN надає можливості підвищення кваліфікації й освоєння нових областей. Наприклад, якщо електрик може приєднати до убудованого модуля тільки маленький температурний сенсор розміром 10×20 мм, те йому не важко буде розібратися в технології регулювання температури й освоїти область створення штучного мікроклімату.

Принцип функціонування шини LCN

Шина LCN є командно-орієнтованою системою. Відправлена клавішею телеграма-команда повністю описує виконавчу функцію. Ця команда могла б звучати, наприклад, у такий спосіб: «активізувати другий вихід через 5 секунд на 90% освітленості».

Цей процес відрізняється від колишньої практики: програмування функції в активаторі й вимоги її виконання за допомогою подачі команди, наприклад перемиканням клавіші. LCN використовує для передачі даних мінімум 24 біта

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

корисної інформації, що перебуває в сенсорі, а не в активаторі. Ця відносно висока маса даних приводить до більшої гнучкості програмування.

Таким чином, 10 різних клавіш відповідними символами описують 10 різних часів, збільшення й зниження яскравості освітлення або зміни світлових сцен. Це відбувається завдяки простому збереженню команди у встановленому клавішному вимикачі.

Зв'язок з іншими системами

Сполучення системи LCN з іншими системами легко здійснено завдяки командно-орієнтованій роботі.

Установники обох шин не повинні домовлятися, щоб призначити функцію, наприклад, для активатора. Кожний може розпорядитися відповідним активатором по власному розсуді. Необхідно тільки знати, які є модулі в системі LCN. Цю інформацію можна одержати безпосередньо із шини. При цьому монтери одержує не тільки інформацію про функції відповідного приладу, але й точний опис завдань модулів у текстовій формі.

Повідомлення про поточний робочий стан можна одержати із шини в будь-який момент. Тому система LCN сумісна з іншими системами.

Не тільки в об'єктах цільового призначення, але й в Smart Home бажано постійно або по запиті мати подання про стан роботи сенсорів і активаторів. У цьому випадку шинна система має незаперечну перевагу перед звичайною кабельною системою. У будь-якому місці шини можуть бути встановлені табло або так звана Home-Management-System. Сигнальне проведення, необхідний у випадку звичайного кабельного з'єднання, є тут зайвим. Це забезпечує скорочення інсталяційних витрат, а разом з тим і вартості системи.

Також важливі ролі грають механізми індикації статусу й квітування. У випадку деяких систем створюється враження, що про це просто забули. Іноді пропонується спрощений метод контролю: сигнальна лампочка показує, що ввімкнувся активатор. Чи відбулося це насправді, залишається невідомим.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

LCN має у своєму розпорядженні триступінчасту систему індикації статусу системи: функціональне квітування, індикація статусу, команди статусу.

1. Функціональне квітування. Необхідно запитувати функціональне квітування для кожної телеграми. У ході цього процесу повинне виконуватися квітування не тільки прийому повідомлень, але й виконання заданих функцій. Наприклад, прийнята команда «опустити жалюзі». Через застопорення механізму або іншого робочого стану вона не може бути виконана. У такому випадку не досить відіслати назад квитанцію «команда прийнята». Система LCN посилає службове повідомлення «команда не виконана» і, більше того, указує причину відмови. Подальшим важливим аспектом є безпосередній зв'язок передавача із приймачем і той факт, що LCN-приймач може самостійно створювати телеграми квітування. Якщо в проміжках включені маршрутизатори або з'єднувачі, то вони автоматично передають функціональну квитанцію далі. Причина, по якій механізм квітування не функціонує в системі EIB, полягає в тому, що формування квитанції відбувається вже на першому з'єднувачі, хоча приймач ще взагалі не одержав ніякої команди.

2. Індикація статусу. Статус всіх входів і виходів модулів системи LCN постійно перевіряється спеціальною програмою-монітором. Тому кожна команда, кожний вхідний сигнал, а також сигнал про несправність моментально надходять у шину. LCN має у своєму розпорядженні спеціальний канал повідомлення, що видає інформацію про кожне повідомлення в режимі реального часу від місця виникнення події. Світлові табло або Home-Management-System постійно опитують канал. Внаслідок цього налагодження режиму візуалізації не викличе в майстра ніяких складностей. Майстер повинен вибрати зі списку всіх модулів модуль індикації й визначити параметри відображення: статус входів, виходів, реле або показання інших табло або установок візуалізації. Стан лампочок на світловому табло може підтримувати чотири режими: «вмк/вимк/мигає» і «мерехтить». Лампочка може мигати, якщо в групі світильників який-небудь із них не працює. При наявності системи Home-Management-System можливо навіть

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

визначення точного місця розташування неробочого світильника. Також можна створювати запит на те, щоб одночасно відображалися перше й останнє значиме повідомлення. За допомогою реєстра сум можна зв'язувати й обробляти повідомлення. Обшир функцією формування повідомлення керує модуль системи. У якості табло використовується 4+1-діодна стандартна клавішна панель EIB.

3. Команди статусу. Команда статусу – це третій щабель системи індикації LCN. Вона дає можливість компоненту відправляти команду на шину завжди, коли що-небудь діє на вході або виході модуля. Наприклад, вихід модуля, відповідальний за освітлення, міг би також управляти вентилятором у ванню. Тому що працюючий вентилятор створює сильний шум, то під час користування душем можна змусити його працювати тільки на 25%. Після того як людина покине ванну кімнату й виключить світло, вентилятор увімкнеться на повну потужність і зробить ґрунтовне очищення повітря. Для цього необхідно виконати тільки дві LCN-команди.

Інсталяція системи LCN не викличе в електрика ніяких труднощів, тому що вона не вимагає попередніх робіт, що підготовляють будинок до установки сучасного встаткування. Наявність вільної жили кабелю NYM є достатньою умовою для початку установки системи. Система не має потреби в блоці живлення, тому витрати на її інсталяцію значно знижуються. Завдяки великій кількості функцій, які можуть одночасно виконувати модулі, потрібна менша кількість спеціальних компонентів. Висока потужність шини LCN забезпечує навіть роботу великого встаткування. LCN пересилає не тільки окремі біти, а докладні функціональні описи. Це забезпечує у свою чергу високу гнучкість у параметруванні встаткування. Для навчання параметруванню, згідно даним виробника, досить одного дня.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Спеціалізоване програмне забезпечення (ПЗ) на комп'ютері загального призначення можна розділити на кілька груп:

– Функціонально закінчене ПЗ, орієнтоване на виконання конкретних споживчих завдань. Прикладом може служити програма Home Assistant компанії Siemens. У рамках цієї програми вже створені користувацькі інтерфейси для керування освітленням, опаленням, електроспоживанням й іншим устаткуванням. Залишається тільки настроїти програму, тобто прив'язати її до реальних датчиків і виконавчих механізмів. Такі програми мають убудовану підтримку EIB-протоколу й забезпечують зв'язок з інсталяцією по послідовному інтерфейсу RS-232 та USB.

– Інструментарій для створення користувацьких інтерфейсів. Багато постачальників EIB компонент (Siemens, Merten, ABB, Gira і ін.) пропонують так звані програми візуалізації, які дозволяють інсталятору самостійно формувати користувацький інтерфейс із бібліотеки елементів і зв'язувати їх із установленим устаткуванням. Як і попередня група програм, вони також мають убудовану підтримку EIB-протоколу.

– Системне програмне забезпечення, що забезпечує доступ до об'єктом EIB-інсталяції з інших програм з підтримкою сучасних стандартів обміну даними по локальних і глобальних комп'ютерних мережах. Гарним прикладом такого ПЗ служить бібліотека функцій Falcon, розповсюджувана Асоціацією EIB. Ця 32-розрядна бібліотека побудована на основі технології DCOM для операційних систем Windows XP/7 і забезпечує для сторонніх програм можливість доступу до EIB-пристроїв і їхніх комунікаційних об'єктів, до групових адрес, а також багатий сервіс по тестуванню системи. Асоціація EIB також поширює побудований на базі цієї бібліотеки OPC-сервер, що забезпечує для існуючих на ринку програмних комплексів керування спорудженнями стандартний інтерфейс із EIB-інсталяцією.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Унікальність систем автоматизації будинків LCN полягає в ряді переваг, які дозволяють їм упевнено конкурувати із системами інших виробників, заснованих на EIB/KNX устаткуванні:

- спрощений монтаж (для передачі даних використовується всього одне резервне проведення в стандартному NYM-електрокабелі);
- проста проєктировка;
- технологічно більше надійний спосіб передачі інформації (дозволяє використовувати систему в умовах підвищеної чутливості до електроперешкод);
- більш низькі витрати на установку (для системи не потрібно прокладати окрему мережу й забезпечувати її живлення);
- можливість застосування всіх типів кнопок і спеціальних настінних контролерів від сторонніх виробників (ABB, Gira, Merten і ін.).

Функціонал устаткування LCN:

- охоронно-пожежні системи, системи захисту від повеней і від зловмисників (датчики, сенсори, сигналізації, блокування дверей і вікон, виклик міліції й т.п.);
- контрольна система «свій – чужий» (ідентифікаційні карти, чипи, біометричний контроль, відеоспостереження, імітація присутності людини й т.п.);
- мануальне й/або програмувальне керування електроприводами (жалюзі, воріт, вікон і т.п.);
- віддалене керування будь-якими компонентами системи (SMS, UMTS, GPRS);
- керування освітленням і електроживленням («розумна» енергозберігаюча система електроживлення, автоматизація й сценарії включення й вимикання світла й електроприладів, різні режими освітлення, централізоване керування й т.п.);
- керування опаленням, вентиляцією й кондиціонуванням (очищення повітря, підтримка особливого клімату, керування теплими підлогами, диференційована клімат у різних приміщеннях або для різних користувачів);

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– керування мультимедійними пристроями (домашні кінотеатри, музичні системи, сценарне підсвічування) і інше.

LCN інтелектуальні bus-модулі. Інтелектуальні bus-модулі – це універсальні компоненти LCN системи, кожне з яких має мікропроцесор і розрізняється функціоналом і областями застосування:

– керування світлом (на театральному рівні, до 200 світлових сцен і ефектів);

– керування освітленням (у тому числі, залежно від денного світла й погоди);

– часове програмування;

– керування вентиляцією;

– керування опаленням;

– керування електроживленням;

– керування електроприводами \ електромоторами (жалюзі, ворота, рольставні й ін.);

– контроль аварійних ситуацій;

– контроль відвідувачів (у тому числі ідентифікація);

– керування системами безпеки декількох рівнів зі складними умовами (у тому числі, блокування замків, попередня й основна тривога, керування декількома сенсорами й акторами й т.п.).

LCN перемикачі, сенсори, модулі. У даній групі компонентів системи LCN представлені:

– бінарні сенсори;

– пристрої відстеження стану до 8 контактів;

– фільтри перешкод;

– функ-приймачі;

– адаптер-кабелі;

– адаптери для з'єднання з різним перемикачами (у тому числі багатоклавішними, EIB/KNX і т.п.);

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– адаптери для перемикачів різного типу.

LCN адаптери, перехідні пристрої. У даній групі компонентів системи LCN представлені:

- розподільні підсилювачі;
- адаптери для оптоволоконного кабелю;
- пасивні стикувальні модулі для з'єднання bus-системи з комп'ютером;
- з'єднувач bus-системи.

LCN реле, виходи:

- реле;
- релейні блоки;
- інтерфейси;
- фільтри перешкод і ін.

LCN сенсори, датчики:

- повні сейсмологічні станції;
- перетворювачі аналогового сигналу;
- датчики руху;
- вітряні сенсори (у тому числі з інтегрованими лічильниками імпульсів);
- світлові сенсори для внутрішнього освітлення;
- сенсори освітленості для зовнішнього застосування;
- сенсори дощу й роси (у тому числі з убудованими адаптерами);
- цифрові температурні датчики й ін.

LCN транспондери, аксесуари та інше:

- електро-термічний сервопривод для вентиляції радіатора батареї;
- випрямляч, що згладжує перешкоди;
- пристрій перетворення напруги в мережі в потенційно вільний контакт;
- touch-панель;
- мультиплікатор і подовжувач I-порту;
- 3-х полярний контактний блок із клемми;
- табло з 8-ю клавішами й 12-ю LED;

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- навчальний кофр, валіза з LCN bus-системою;
- низьковольтний блок живлення;
- sms-модуль.

LCN сенсорні панелі керування. Елегантні дизайн-вимикачі з мінерального скла. Комфорт, добірність і багатофункціональність:

- 4, 6 і 12 клавiш із 12, 18 і 36-ю функціями відповідно;
- величина й форма клавiш програмувальна (тобто настроюється на будь-який смак);
- індивідуальний дизайн клавiш керування завдяки змінному вкладишу за склом;
- 6 і 12 LED-індикаторів відображаючих статус;
- підсвічування клавiш зсередини;
- GT12: LED температурна гістограма;
- GT6D і GT4D: брильянтова якість OLED дисплея;
- інтегрований ІЧ-приймач для пульта.

LCN контроль доступу, дистанційне керування:

- транспондери;
- пристрої, що зчитують;
- приймач інфрачервоного випромінювання;
- пульт дистанційного керування (у тому числі пульти великої дальності);
- універсальні транспондери-пристрої, що зчитують;
- антена транспондерної системи;
- пластикова карта;
- транспондерні брелоки.

3.2 Розробка структурної схеми

На рисунку 3.1 зображена узагальнена структурна схема інтелектуального дому.

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.

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.

- 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.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– 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.

Інтелектуальний будинок дозволяє замінити всі пульти управління однією або декількома (по кількості зон або кімнат) сенсорними панелями. Вони дозволяють не ламати голову над тим, як підбудувати середовище перебування під необхідні умови.

Використання сучасного устаткування дозволяє створити в будинку єдиний комплекс із систем безпеки (охоронна й пожежна сигналізація, відеоспостереження, контроль доступу), систем зв'язку й комунікації (телефонна й комп'ютерна мережі, оповіщення, екстрений виклик), систем управління опаленням, вентиляцією, освітленням і т.д., що працює по обраному алгоритму.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

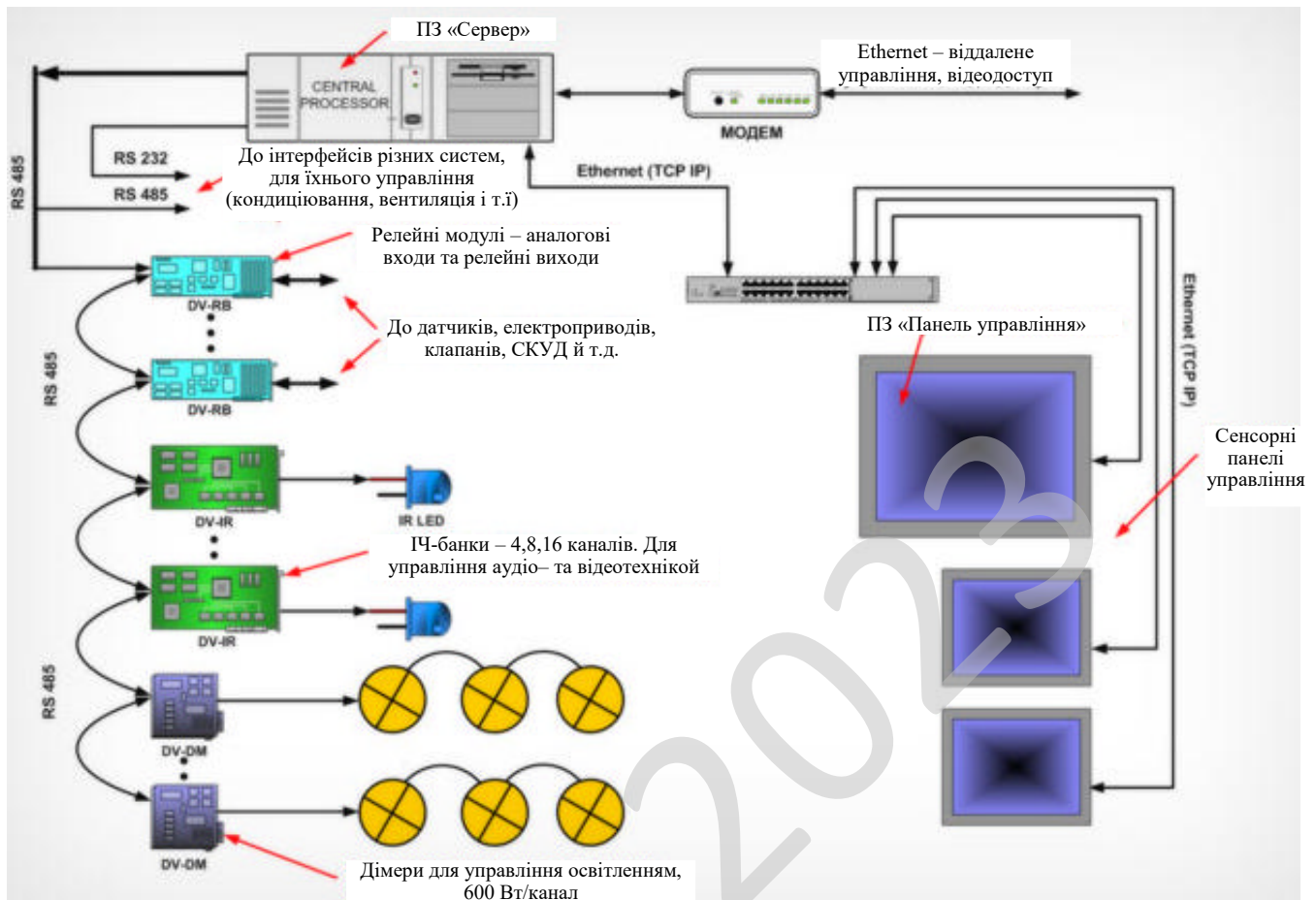


Рисунок 3.1 – Структурна схема системи

Як ілюстрацію можна привести приклад, коли автоматизація й включення в єдиний контур управління освітлювальної системи й систем клімат-контролю (опалення, кондиціонування й вентиляція) будинку (окремої квартири) дозволяють реалізувати автоматичне управління цими системами залежно від пори року й доби, умов навколишнього середовища, присутності людей і інших факторів. У результаті досягається істотне зниження витрат на електроенергію й теплопостачання. Досвід показує, що економія експлуатаційних витрат у цьому випадку може досягати 15-20%.

Состав системи:

– Центральний процесор. Обробка керуючих команд від керуючих сенсорних панелей, відправлення команд для ІЧ банків, диммерів, релейних

модулів. Обробка вхідної інформації від датчиків, контролерів, модемів. Установлюється програмне забезпечення розроблене у результаті виконання магістерського проектування, його серверна частина.

– Керуючі сенсорний панелі. Передача в центральний процесор команд керування, відеомоніторинг, подання на екрані всієї необхідної інформації/пов'язані із центральним процесором по мережі Ethernet (TCP/IP)

– Діммери. Регулювання яскравості світіння ламп, управляються центральним процесором або по командах від сенсорних панелей або відповідно до закладеного алгоритму (сценарію), пов'язані із центральним процесором RS-485 інтерфейсом.

– Релейні модулі. Керування релейними контактами й прийом сигналів на входи типу 0-1 управляються Центральним процесором або по командах від Керуючих сенсорних панелей або відповідно до закладеного алгоритму(сценарієм)/пов'язані із Центральним процесором RS-485 інтерфейсом.

– Банки ІЧ команд. Випромінювання ІЧ команд із пам'яті контролера, ранне туди записаних. Керування апаратурою що має ІЧ інтерфейс. Управляються центральним процесором або по командах від керуючих сенсорних панелей або відповідно до закладеного алгоритму/пов'язані із центральним процесором RS-485 інтерфейсом.

Отже, для того щоб зробити розумний будинок потрібно:

– Зробити схему розташування периферійного встаткування на об'єкті, робиться на поетажних планах.

– Пронумерувати всі приміщення в будинку.

– Скласти список проводів/тип кабелю, маркування, звідки й куди йде проведення, які повинні підходити до кожному елементу системи кабельний журнал.

– Відкрити вікно “Редактор інтелектуального дому” і мишкою створити необхідні елементи системи спочатку сторінки кімнат, потім на цих сторінках і інші елементи керування будинком. Під час роботи в редакторі ви створюєте

						ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			38

графічний інтерфейс керування системи. Наприклад, створивши кнопку на сторінці “Вітальня” ви можете вибрати в редакторі ряд дій, які повинні відбутися при її натисканні, ну скажемо, відправити ІЧ команду на включення кондиціонера на температуру 23С цю команду при програмуванні системи потрібно буде записати в ІЧ банк або включити освітлення в акваріумі в рибок тобто вибрати запропонований диммер і вказати в редакторі на яку яскравість варто включити лампу.

– Тепер залишилося додати графічному інтерфейсу індивідуальний дизайн і підключити до центрального процесора необхідне встаткування/список цього встаткування також формується в процесі роботи редактора.

Основні складові інтелектуального будинку

Зв'язок:

- телефонний зв'язок внутрішня (інтерком) і локальні АТМ;
- телефонний зв'язок зовнішня провідна, радіорелейна, супутникова;
- інтернет–телефонія;
- системи відеоконференцій.

Мультимедіа:

- наземне й кабельне телебачення;
- супутникове телебачення;
- домашнє відео;
- домашній кінотеатр;
- мульти– аудіо й відео (multiroom).
- один пульт управління для всіх систем

Інформаційні системи:

- локальна комп'ютерна мережа;
- домашній офіс;
- широкополосний доступ у глобальну мережу (Інтернет).

Безпека:

- пожежна сигналізація й система автоматичного пожежогасіння;

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- охоронна сигналізація;
- система контролю доступу;
- система зовнішнього й внутрішнього відео спостереження;
- система управління паркінгом;
- система внутрішнього оповіщення (радіомережа);
- аварійний контроль інженерних систем;
- система екологічного контролю.

Водопостачання й газопостачання:

- автоматичне наповнення ванн, басейнів і накопичувальних резервуарів;
- автономне й резервне нагрівання води;
- запобігання витоків водопроводу й витоків газу;
- управління ландшафтними водними системами (фонтани, водоспади);
- управління температурою води у ваннах і басейнах (термостатування);
- облік витрат й управління споживанням води й/або газу.

Система електроживлення:

- безперебійне й гарантоване електропостачання;
- захист від поразки електричним струмом людей і тварин;
- автоматизація управління електроживленням побутових приладів;
- запобігання перевантажень і короткого замикання в електричній мережі;
- управління якістю електроживлення – моніторинг, стабілізація й фільтрація;
- облік витрат й управління споживанням електроенергії.

Освітлення внутрішнє й зовнішнє:

- аварійне й чергове освітлення;
- автоматизація управління світлом;
- гнучке настроювання світлових груп;
- дистанційне й віддалене управління світлом;
- програмування світлових сцен.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Опалення, вентиляція, кондиціонування:

- автоматизація управління температурою повітря;
- контроль якості повітря;
- узгодження роботи різних кліматичних систем;
- облік витрати й управління споживанням теплової енергії.

Тепер спробуємо створити зі звичайного будинку інтелектуальний. Для цього нам потрібно всі прилади, які виконують перераховані вище функції, об'єднати в одну систему й підключити її до віддаленого сервера. Тим самим буде управління кожним компонентом, та всіма воедино. І необхідно щоб контроль за станом всіх приладів користувач міг одержати в будь-який момент часу, навіть перебуваючи поза будинком. Для цього треба організувати мережу в будинку, об'єднавши всі прилади й системи над якими треба здійснити контроль, та потім вибрати спосіб підключення до віддаленого сервера.

3.3 Розробка функціональної схеми

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

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

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

- 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 автоматизацію, що дозволяють

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

незалежно від людини виконувати роботи по підтримці дому у належному стані, тобто сервер автоматизації приймає рішення включати той або інший прибор.

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

- управління з екрану ЕОМ;
- релейне управління;
- диммування;
- «проходні» схеми вимикання;
- бездротове управління електрикою;
- для аудіо/відео та світла один пульт ДУ;
- віддалене управління за телефоном.

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 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».

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

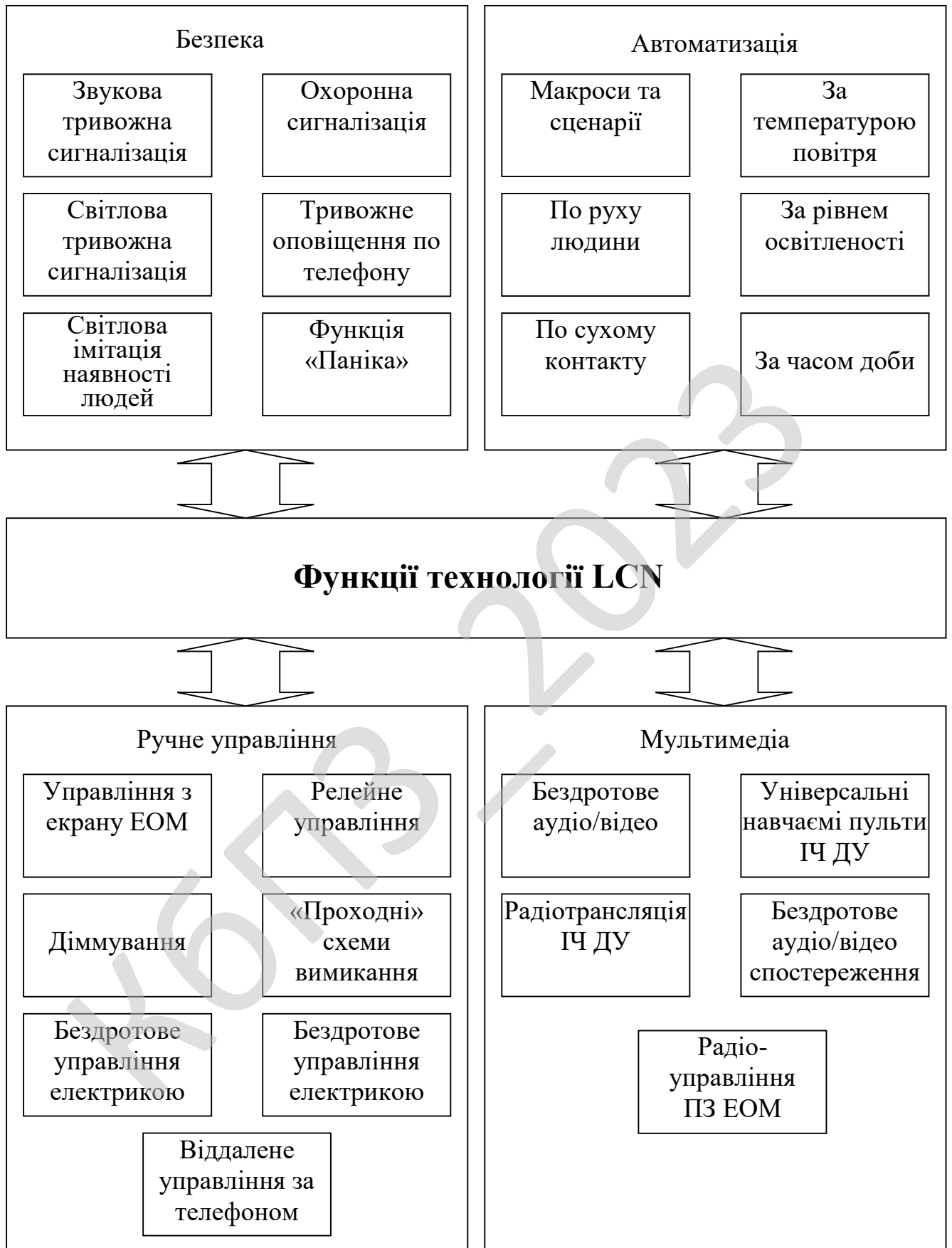


Рисунок 3.2 – Функціональна схема системи

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

- макроси та сценарії;
- за температурою повітря;
- по руху 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 взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

Першим процесом, який завантажується у системі є процес ввімкнення системи LCN.

Він взаємодіє з наступними процесами:

- Процес керування пристроями будинку.
- Процес читання показів пристроїв.

Процес керування пристроями будинку взаємодіє з наступними процесами:

- Процес керування освітленням.
- Процес керування системою водопостачання.
- Процес керування телефоном.
- Процес керування домашнім кінотеатром.
- Процес керування системою безпеки.
- Процес керування кліматом.
- Процес керування системою водопостачання.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44



Рисунок 3.3 – Діаграма процесів системи

Процес керування системою безпеки взаємодіє з наступними процесами:

- Процес блокування/відкриття вхідних дверей.
- Процес охорони та пожежної сигналізації.
- Процес відеоспостереження.

Процес керування кліматом взаємодіє з наступними процесами:

- Процес керування опаленням.
- Процес керування кондиціонером.

Процес читання показів пристроїв взаємодіє з наступними процесами:

- Процес показу температури.
- Процес показу вологості.
- Процес поточного стану пристроїв.
- Процес повідомлення системи безпеки.

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

4.1 Блок–схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків.

Спершу відбувається виведення основного вікна програми. Після цього запускається ряд вкладень, які дозволяють керувати тим, або іншим елементом інтелектуального дому.

Якщо обрано управління освітленням, тоді виконуються наступні дії:

- Визначається поточний стан освітлення.
- Виводиться поточний стан освітлення.
- Відбувається керування освітленням та його яскравістю.

Якщо обрано управління водопостачанням, тоді виконуються наступні дії:

- Визначається поточний стан водопостачання.
- Виводиться поточний стан водопостачання.
- Відбувається управління системою контролю протікання води, клапанами холодної та гарячої води.

Якщо обрано управління газопостачанням, тоді виконуються наступні дії:

- Визначається поточний стан газопостачання.
- Виводиться поточний стан газопостачання.
- Відбувається управління системою витoku газу та системою виявлення диму.

Якщо обрано управління кліматом, тоді виконуються наступні дії:

- Визначається поточний стан температури та вологості.
- Виводиться поточний стан температури та вологості.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

– Відбувається управління опаленням, кондиціонерами та жалюзі на вікнах.

Якщо обрано управління телефонним зв'язком, тоді виконуються наступні дії:

- Визначається поточний стан телефонного зв'язку.
- Виводиться поточний стан телефонного зв'язку.
- Відбувається управління телефоном та автовідповідачем.

Якщо обрано управління домашнім кінотеатром, тоді виконуються наступні дії:

- Відбувається управління домашнім кінотеатром.

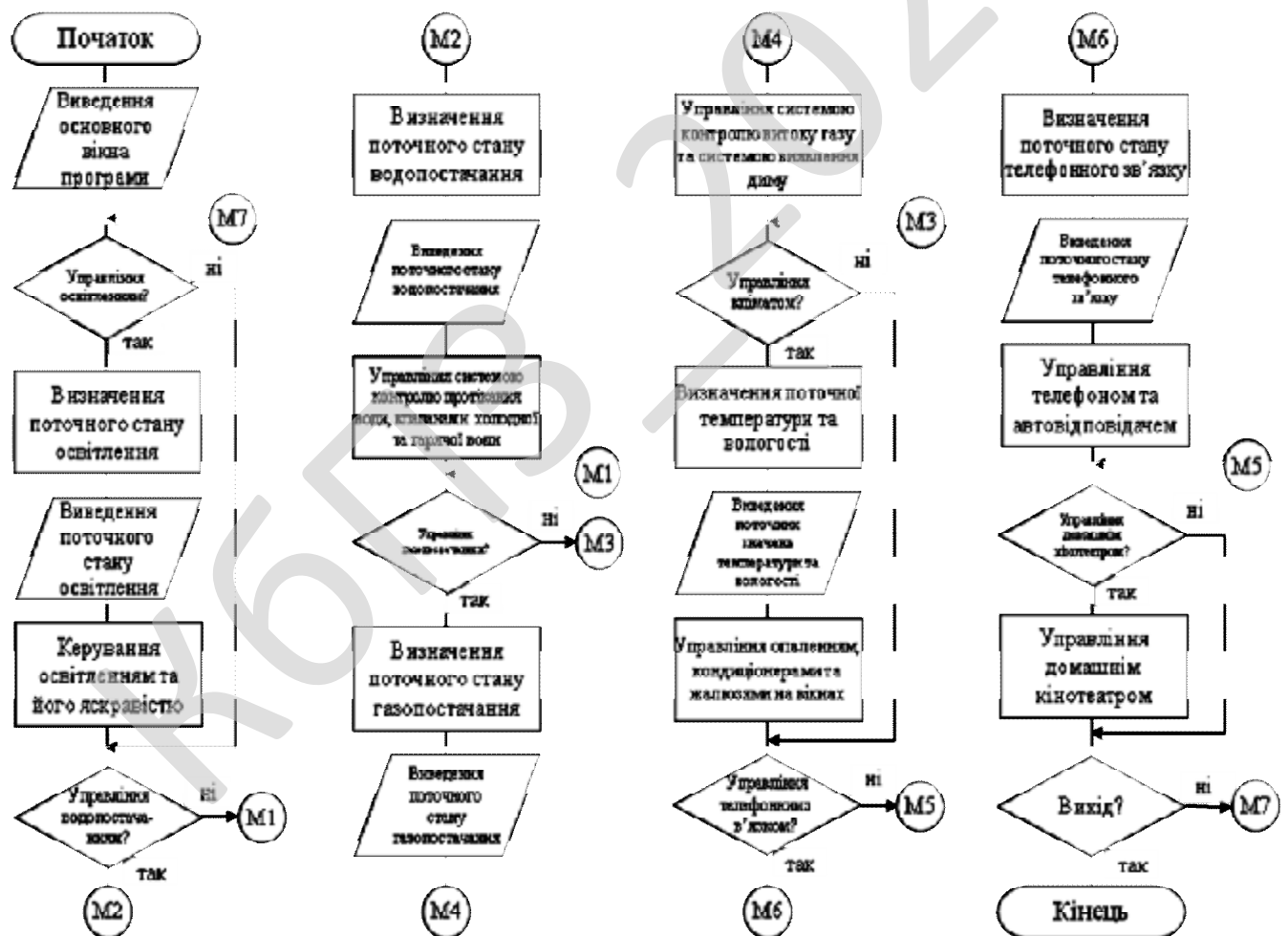


Рисунок 4.1 – Блок-схема роботи основної програми

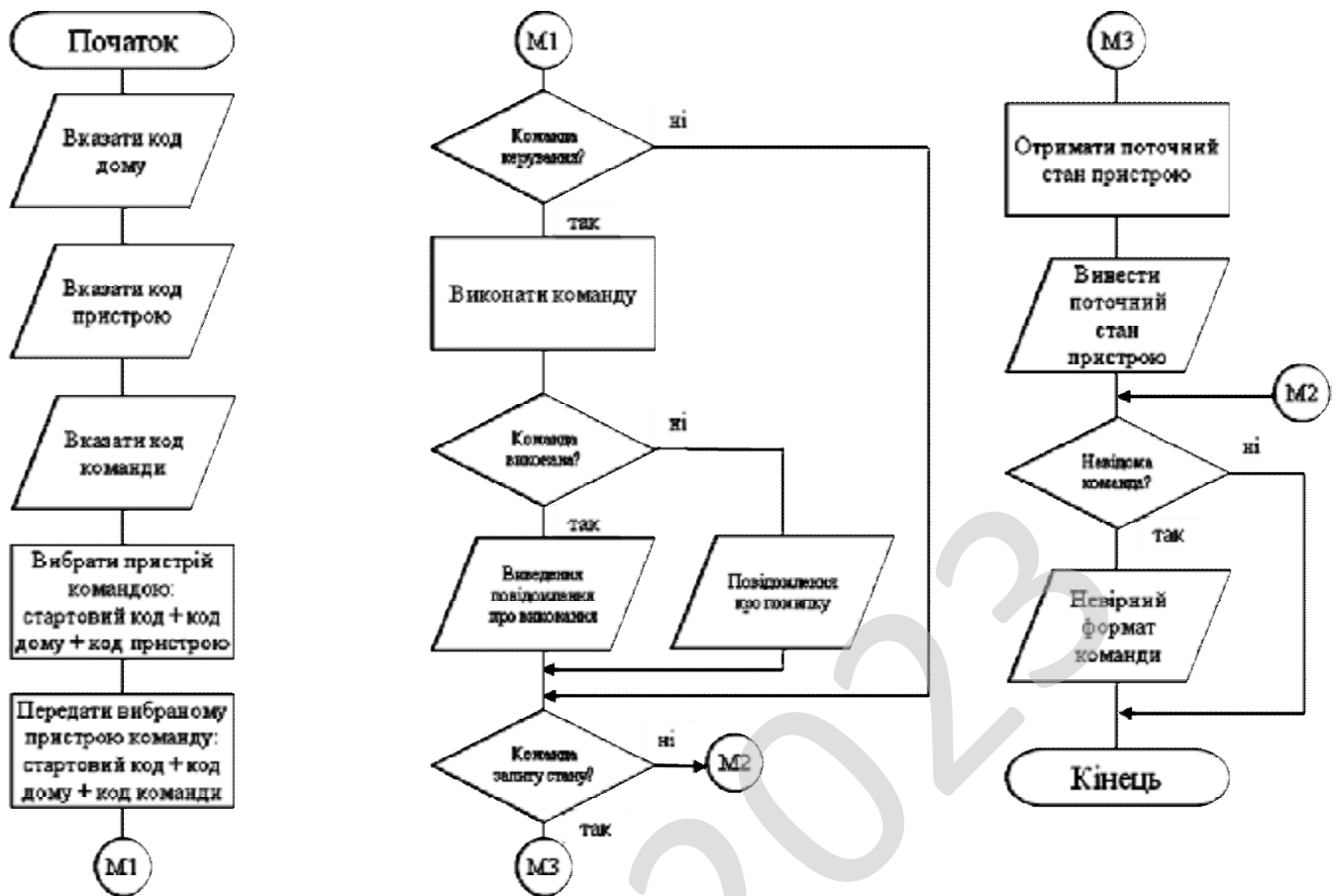


Рисунок 4.2 – Блок-схема роботи підпрограми управління пристроями інтелектуального будинку

На рисунку 4.2 зображено блок-схему роботи підпрограми управління пристроями інтелектуального будинку. Вона працює наступним чином.

Спершу послідовно вказується код дому, код пристрою, код команди.

Після цього обирається пристрій командою: стартовий код + код дому + код пристрою.

Передається обраному пристрою команда: стартовий код + код дому + код пристрою.

Далі визначається тип команди.

Якщо команда керування, тоді виконуються наступні дії:

– Виконується команда.

– Якщо команда виконалася, тоді виводиться повідомлення про

виконання. У іншому випадку виводиться повідомлення про помилку.

Якщо запускається команда запиту стану, тоді виконуються наступні дії:

- Отримується поточний стан пристрою.
- Виводиться поточний стан пристрою.

Якщо ж команда невідома, тоді виводиться повідомлення про те, що невірний формат команди.

Наведемо код головної програми.

```
USEFORM("main.cpp", Form_main); // Головне вікно
USEFORM("about.cpp", Form_about); // Вікно даних про розробника
USEFORM("light.cpp", Form_light); // Вікно управління світлом
USEFORM("water.cpp", Form_water); // Вікно управління водою
USEFORM("gas.cpp", Form_gas); // Вікно управління газом
USEFORM("security.cpp", Form_security); // Вікно управління захистними системами
USEFORM("climate.cpp", Form_climate); // Вікно управління кліматом
USEFORM("phone.cpp", Form_phone); // Вікно управління телефонією
USEFORM("tv.cpp", Form_tv); // Вікно управління телевізорами
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        // Створення головної форми
        Application->CreateForm(__classid(TForm_main), &Form_main);
        // Створення форми даних про розробника
        Application->CreateForm(__classid(TForm_about), &Form_about);
        // Створення форми управління світлом
        Application->CreateForm(__classid(TForm_light), &Form_light);
        // Створення форми управління водою
        Application->CreateForm(__classid(TForm_water), &Form_water);
        // Створення форми управління газом
        Application->CreateForm(__classid(TForm_gas), &Form_gas);
        // Створення форми управління захистними системами
        Application->CreateForm(__classid(TForm_security), &Form_security);
        // Створення форми управління кліматом
        Application->CreateForm(__classid(TForm_climate), &Form_climate);
        // Створення форми управління телефонією
        Application->CreateForm(__classid(TForm_phone), &Form_phone);
        // Створення форми управління телевізорами
```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

        Application->CreateForm(__classid(TForm_tv), &Form_tv);
        // Сворення форми запуску проекту
        Application->Run();

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

//відкриття вікна "Управління 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)
{

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50


```

//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_LCN_Command(0,1,19,1); //0-код будинку; 1-код пристрою, що керує
ліхтарем; 19-код команди "off"; 1-кількість повторних надсилянть команди
}
//-----
//ввимкнення світла в коридорі з вказаною яскравістю
void __fastcall TForm_light::corridor_onClick(TObject *Sender)
{
corridor->Caption="Ввимкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
TrackBar_corridor->Enabled=true;
ClientSocket->Send_LCN_Command(0,2,18,1);
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_corridor->Position);
}
//-----
//ввимкнення світла у вітальні з вказаною яскравістю
void __fastcall TForm_light::drawing_room_onClick(TObject *Sender)
{
drawing_room->Caption="Ввимкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
TrackBar_drawing_room->Enabled=true;
ClientSocket->Send_LCN_Command(0,3,18,1);
ClientSocket->SendLeviton_LCN_(0,3,TrackBar_drawing_room->Position);
}
//-----
//ввимкнення світла на кухні з вказаною яскравістю
void __fastcall TForm_light::kitchen_onClick(TObject *Sender)
{
kitchen->Caption="Ввимкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;
ClientSocket->Send_LCN_Command(0,4,18,1);
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,5,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,6,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,7,18,1);
ClientSocket->SendLeviton_LCN_(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;
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

TrackBar_bathroom->Enabled=true;
ClientSocket->Send_LCN_Command(0,8,18,1);
ClientSocket->SendLeviton_LCN_(0,8,TrackBar_bathroom->Position);
}
//-----
//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_LCN_Command(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_LCN_Command(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_LCN_Command(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;
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

ClientSocket->Send_LCN_Command(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_LCN_Command(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_LCN_Command(0,7,19,1);
}
//-----
//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_LCN_Command(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_LCN_Command(0,1,18,1);
ClientSocket->SendLeviton_LCN_(0,1,TrackBar_torch->Position);
corridor->Caption="Ввімкнено";
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
ClientSocket->Send_LCN_Command(0,2,18,1);
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_corridor->Position);
drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
ClientSocket->Send_LCN_Command(0,3,18,1);
ClientSocket->SendLeviton_LCN_(0,3,TrackBar_drawing_room->Position);
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
ClientSocket->Send_LCN_Command(0,4,18,1);
ClientSocket->SendLeviton_LCN_(0,4,TrackBar_kitchen->Position);
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
ClientSocket->Send_LCN_Command(0,5,18,1);
ClientSocket->SendLeviton_LCN_(0,5,TrackBar_cabinet->Position);
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
ClientSocket->Send_LCN_Command(0,6,18,1);
ClientSocket->SendLeviton_LCN_(0,6,TrackBar_bedroom->Position);
baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
ClientSocket->Send_LCN_Command(0,7,18,1);
ClientSocket->SendLeviton_LCN_(0,7,TrackBar_baby_room->Position);
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
ClientSocket->Send_LCN_Command(0,8,18,1);
ClientSocket->SendLeviton_LCN_(0,8,TrackBar_bathroom->Position);
TrackBar_torch->Enabled=true;
TrackBar_bedroom->Enabled=true;
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;

```

					БКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

}
//-----
//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_LCN_Command(0,1,19,1);
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
ClientSocket->Send_LCN_Command(0,2,19,1);
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
ClientSocket->Send_LCN_Command(0,3,19,1);
kitchen->Caption="Вимкнено";
Image_kitchen_off->Visible=true;
Image_kitchen_on->Visible=false;
ClientSocket->Send_LCN_Command(0,4,19,1);
cabinet->Caption="Вимкнено";
Image_cabinet_off->Visible=true;
Image_cabinet_on->Visible=false;
ClientSocket->Send_LCN_Command(0,5,19,1);
bedroom->Caption="Вимкнено";
Image_bedroom_off->Visible=true;
Image_bedroom_on->Visible=false;
ClientSocket->Send_LCN_Command(0,6,19,1);
baby_room->Caption="Вимкнено";
Image_baby_room_off->Visible=true;
Image_baby_room_on->Visible=false;
ClientSocket->Send_LCN_Command(0,7,19,1);
bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
ClientSocket->Send_LCN_Command(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;

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

```

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)
{
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_LCN_Command(0,x,18,1);
y=random (6)+2;
ClientSocket->Send_LCN_Command(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_LCN_(0,1,TrackBar_torch->Position);
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

```

}
//-----
//зміна яскравості освітлення коридору
void __fastcall TForm_light::TrackBar_corridorChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_corridor->Position);
}
//-----
//зміна яскравості освітлення вітальні
void __fastcall TForm_light::TrackBar_drawing_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_drawing_room->Position);
}
//-----
//зміна яскравості освітлення кухні
void __fastcall TForm_light::TrackBar_kitchenChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_kitchen->Position);
}
//-----
//зміна яскравості освітлення кабінету
void __fastcall TForm_light::TrackBar_cabinetChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_cabinet->Position);
}
//-----
//зміна яскравості освітлення спальні
void __fastcall TForm_light::TrackBar_bedroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_bedroom->Position);
}
//-----
//зміна яскравості освітлення дитячої
void __fastcall TForm_light::TrackBar_baby_roomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_baby_room->Position);
}
//зміна яскравості освітлення ванної
void __fastcall TForm_light::TrackBar_bathroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_bathroom->Position);
}
}

```

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 8845:2019 – алгоритм симетричного потокового перетворення. В основі ДСТУ 8845:2019 лежить класична схема підсумовуючого генератора подібна генератору SNOW 2.0, SNOW 3.0 та SNOW V. В основі ДСТУ 8845:2019 збережені всі базові операції шифрів сімейства SNOW, а раунд шифрування AES замінений на функцію нелінійної підстановки T, що реалізовує перестановку елементів скінченного поля $GF(2^{64})$ за допомогою компонентів національного стандарту симетричного криптоперетворення ДСТУ 7624:2014.

ДСТУ 8845:2019 використовує 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K і забезпечує високий та надвисокий рівень стійкості із врахуванням можливого застосування квантового криптографічного аналізу. При розробці алгоритму ДСТУ 8845:2019 орієнтувалися на сучасні 64-бітні обчислювальні системи, тому розмір слова обрано рівним 8 байт. запису байтів застосовують подання від старшого до молодшого. Генератор ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру схематично приведено у стандарті.

Як впливає з генератора ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру основними компонентами генератору є регістр зсуву з лінійним зворотнім зв'язком та скінчений автомат на базі якого виконується нелінійне перетворення T. Вхідні дані (ключ шифрування K та вектор ініціалізації IV) використовуються для ініціалізації змінної стану $S_i (i \geq 0)$, яка складається із двох компонент до складу яких входять [12]:

– 16 змінних $s^{(i)}$ – комірок регістра зсуву з лінійним зворотнім зв'язком: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, s_{13}^{(i)}, s_{12}^{(i)}, s_{11}^{(i)}, s_{10}^{(i)}, s_9^{(i)}, s_8^{(i)}, s_7^{(i)}, s_6^{(i)}, s_5^{(i)}, s_4^{(i)}, s_3^{(i)}, s_2^{(i)}, s_1^{(i)}, s_0^{(i)})$;

– Два регістри скінченного автомату $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. На виході отримуємо ключовий потік (гамма), який формується з 8-байтних слів Z_i .

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

З рисунку слідує, що відводи регістра зсуву з лінійним оберненим зв'язком побудовані за примітивним над полем $GF(2^{64})$ поліномом: $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, де α є коренем примітивного над полем $GF(2^8)$ поліному $gz(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2 z^5 + \beta^{224} z^4 + \beta^{70}z^3 + \beta^2$. Поле $GF(2^8)$ як і в ДСТУ 7624:2014 побудовано за примітивним на полем $GF(2)$ поліномом $p(y) = x^8 + x^4 + x^3 + x^2 + 1$, а коефіцієнти $g(z)$ подаються через ступінь примітивного елементу β поля $GF(2^8)$, тобто β корінь поліному $p(y)$. Тобто, у нас є вежа полів: $GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024})$, де:

- поле $GF(2^{1024})$ задається відводами зворотного зв'язку як фактор кільце $GF(2^{64})[x]/(f(x))$;

- поле $GF(2^{16424})$ задається як фактор кільце $GF(2^8)[z]/(g(z))$;

- поле $GF(2^{1024})$ задається як фактор кільце $GF(2)[y]/(p(y))$.

З вищезазначеного, слідує що період вихідної послідовності становить 2^{1024} .

Структурно в алгоритмі симетричного потокового перетворення ДСТУ 8845:2019 виділяють три основні функції:

- функція ініціалізації, яка приймає в якості вхідних даних 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K, і виробляє початкове значення змінної стану $S_0 = (s^{(0)}, r^{(0)})$;

- функція наступного стану Next, яка приймає на вхід зміну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє наступне значення змінної стану $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$;

- функція ключового потоку Strm, що приймає на вході змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє на виході 64-бітний ключовий потік Z^i .

Також функція Next може виконуватися в двох режимах, в залежності від способу виконання ітерації, як частина реалізації ініціалізації алгоритму ДСТУ 8845:2019 або як частина функції ключового потоку Strm.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення призначене для управління інтелектуальним будинком за технологією LCN.

Програмно-апаратні вимоги до комп'ютера:

- Загальний обсяг ОЗП: 4 Gb.
- Вільний простір на жорсткому диску: 5003 Мбайт.
- Операційна система Microsoft 10.
- Інтерфейс Ethernet.

Перелік необхідного обладнання:

Обов'язкове обладнання:

– Передавачі – дозволяють передавати спеціальні коди команд у форматі LCN по електромережі. Такими пристроями є: **програмуємі таймери**, що посилають сигнали в потрібний час; **комп'ютерні модулі**, що виконують задані програми по керуванню електроприладами; **датчики** температури, освітленості, руху й ін., які при настанні певних подій посилають відповідні сигнали приймачам.

– Приймачі – приймають команди LCN і виконують їх: включають або виключають світло, регулюють освітленість і т.д. На кожному приймачі є селектори установки його адреси: 16 можливих кодів будинку (A – P) і 16 можливих кодів модуля (1 – 16), тобто всього 256 різних адрес. Кілька приймачів можуть мати однакову адресу, у цьому випадку вони управляються одночасно.

– Вимірювальне устаткування – використовується для вимірювання рівнів корисних сигналів LCN і перешкод в електромережі при виконанні монтажних і пусково-налагоджувальних робіт.

– Мережні пристрої – для об'єднання всіх інших пристроїв з комп'ютером (вита пара, комутатори, концентратори тощо).

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Рекомендоване обладнання:

– Лінійне устаткування – повторювачі/ретранслятори сигналів, фільтри стрибків напруги або струму, протишумові фільтри, блокатори сигналів. Ці пристрої використовуються для підвищення надійності й безвідмовності системи в цілому. Хоча в простих системах можливе досягнення прекрасних результатів і без використання цих засобів, але завжди краще підстрахуватися.

– Трансивери – приймають сигнали від інфрачервоних або радіо-пультів дистанційного керування й передають їх в електромережу, перетворивши у формат LCN.

– Пульти ДУ – забезпечують дистанційне керування пристроями LCN по ІЧ або радіо-каналам.

Функціональні можливості програми:

1. Управління освітленням будинку:

- вмикання/вимикання ламп;
- регулювання яскравості світла від ламп;
- перегляд поточного стану ламп;
- імітація присутності господарів, шляхом почергового вмикання вимикання ламп у різних кімнатах (для відлякування крадіїв).

2. Управління водопостачанням:

- система контролю протікання води;
- відкриття/закриття клапанів холодної та гарячої води.

3. Управління газопостачанням:

- система контролю витoku газу;
- система виявлення диму.

4. Управління кліматом будинку:

- перегляд поточної температури та вологості у будинку;
- управління опаленням (вмикання/вимикання, регулювання рівня температури, автоматична робота у вказаному діапазоні часу);
- управління гарячими полами (вмикання/вимикання, регулювання

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

рівня температури);

– управління кондиціонером (вмикання/вимикання, регулювання рівня температури та вологості);

– управління жалюзями на вікнах.

5. Управління телефонним зв'язком:

– управління телефоном (вмикання/вимикання, регулювання гучності, набір номера);

– управління автовідповідачем (вмикання/вимикання, гучність).

6. Управління домашнім кінотеатром:

– вмикання/вимикання телевізора;

– регулювання гучності;

– вибір телеканалу.

7. Управління системою безпеки будинку:

– управління охоронною сигналізацією;

– управління пожежною сигналізацією;

– управління системою відеоспостереження;

– блокування/відкриття вхідних дверей.

Головне вікно програми зображене на рисунку 5.1.

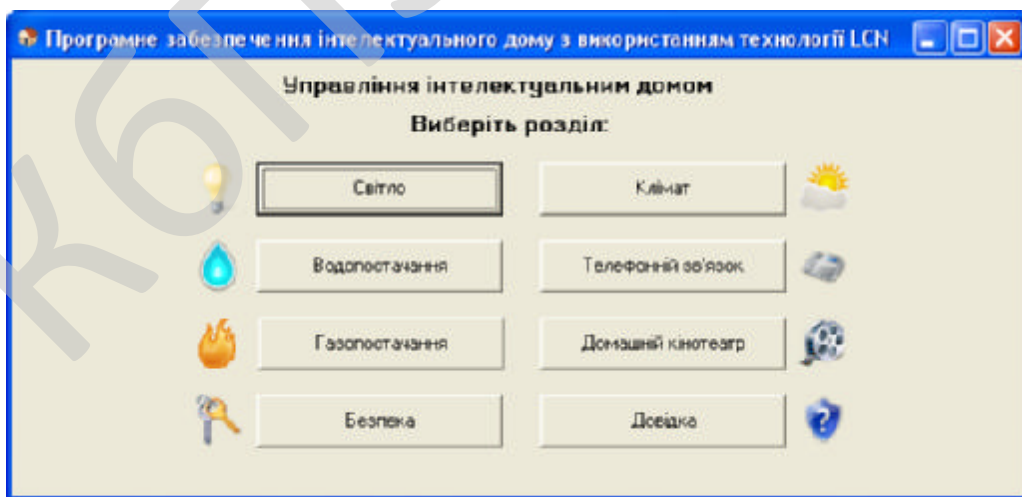


Рисунок 5.1 – Головне вікно програми

З рисунку видно, що за допомогою головного вікна користувач здійснює вибір розділу програми, в який він хоче перейти, а саме: світло, водопостачання, газопостачання, безпека, клімат, телефонний зв'язок, домашній кінотеатр, довідка.

Вікно управління освітленням зображене на рисунку 5.2.

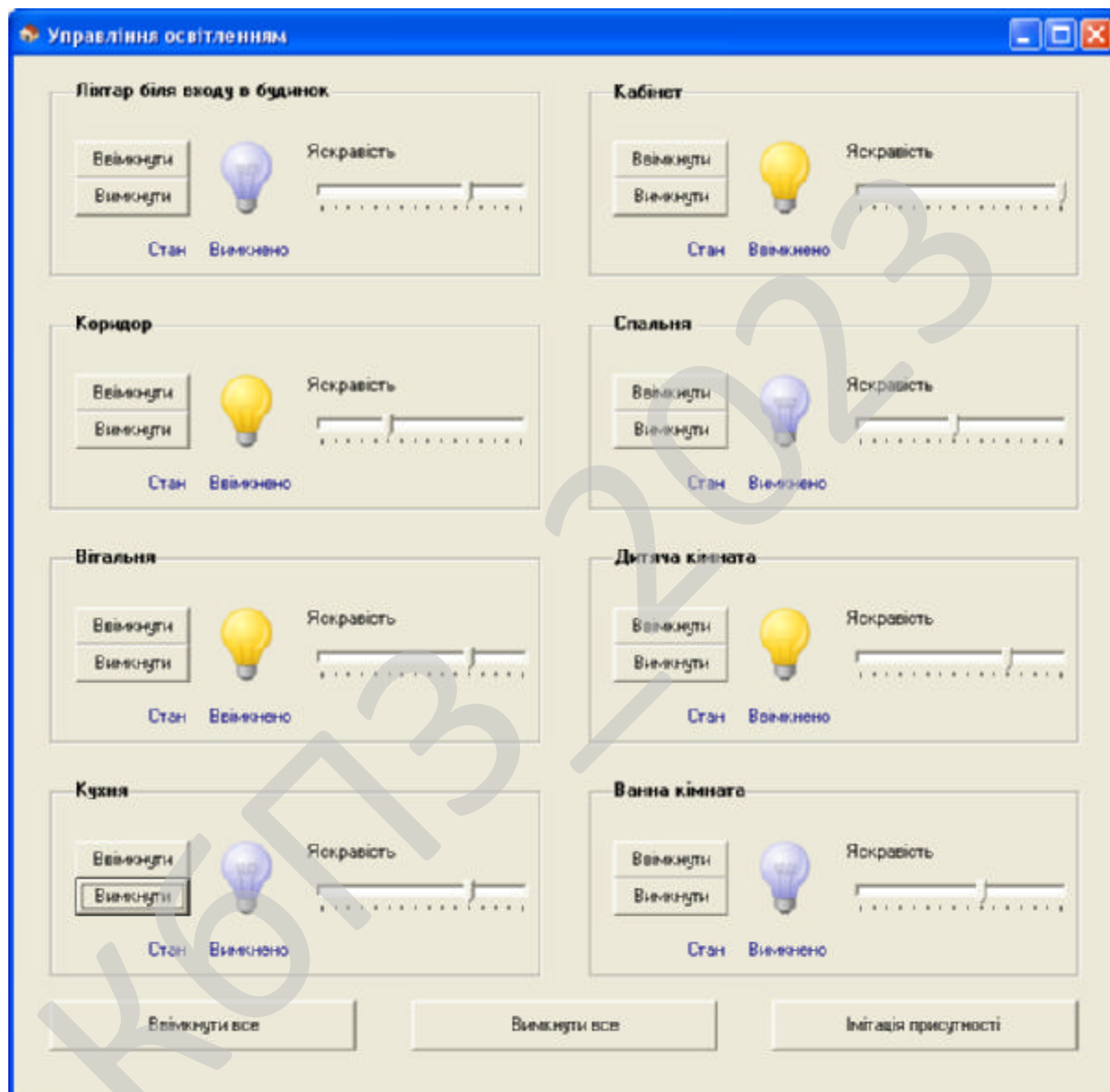


Рисунок 5.2 – Управління освітленням

Вікно управління водопостачанням зображене на рисунку 5.3.

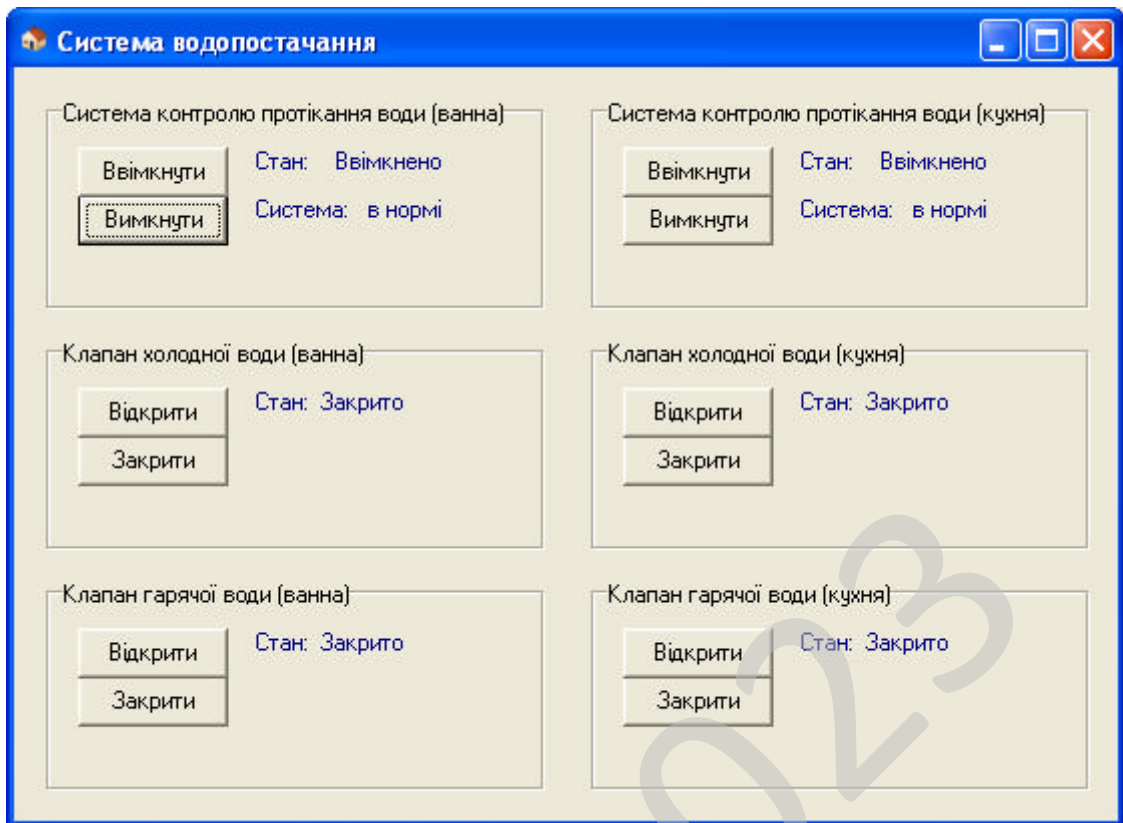


Рисунок 5.3 – Управління водопостачанням

Вікно управління газопостачанням зображене на рисунку 5.4.

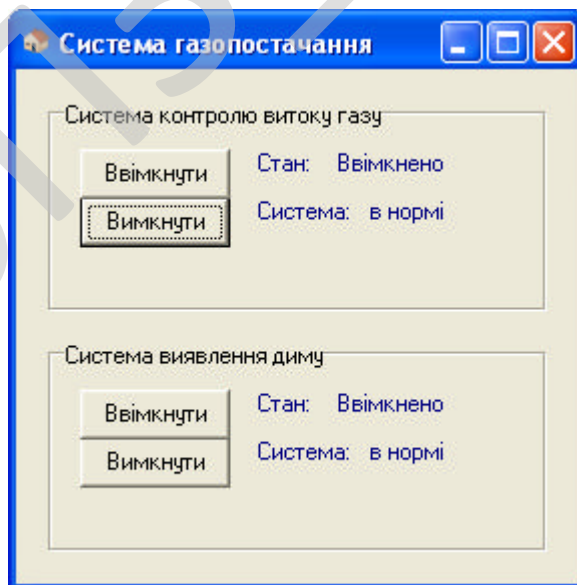


Рисунок 5.4 – Управління газопостачанням

Вікно управління системою безпеки будинку зображене на рисунку 5.5.

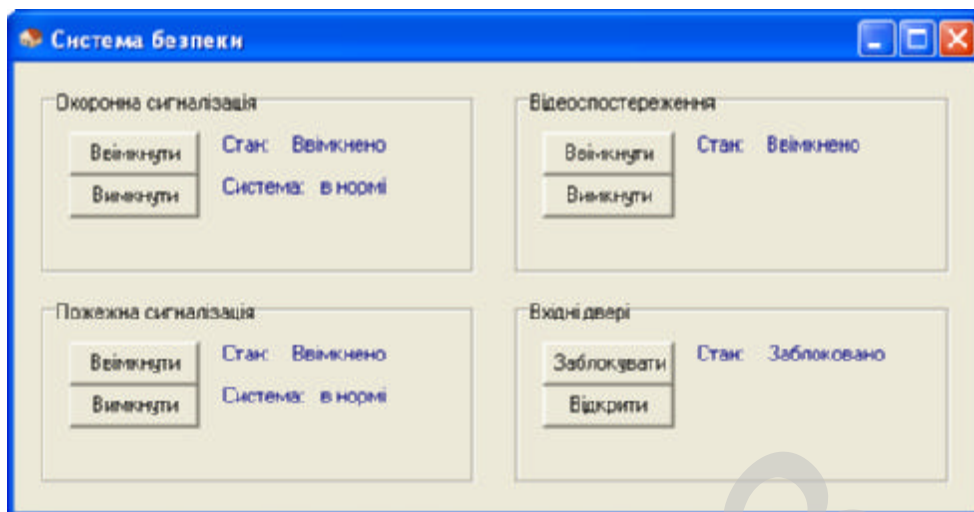


Рисунок 5.5 – Система безпеки будинку

Вікно управління кліматом будинку зображене на рисунку 5.6.

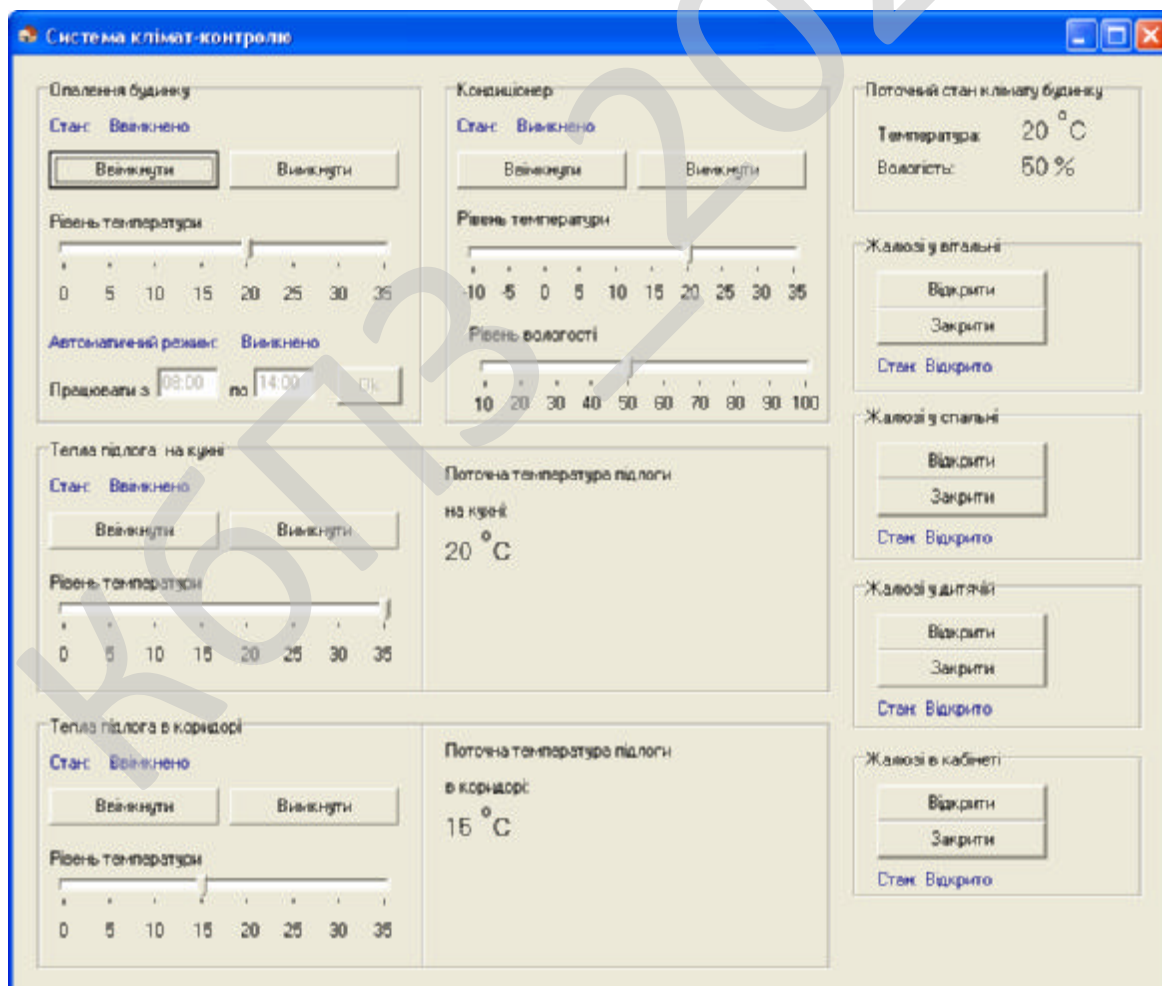


Рисунок 5.6 – Управління кліматом

Вікно управління телефонним зв'язком зображене на рисунку 5.7. Вікно управління домашнім кінотеатром зображене на рисунку 5.8.

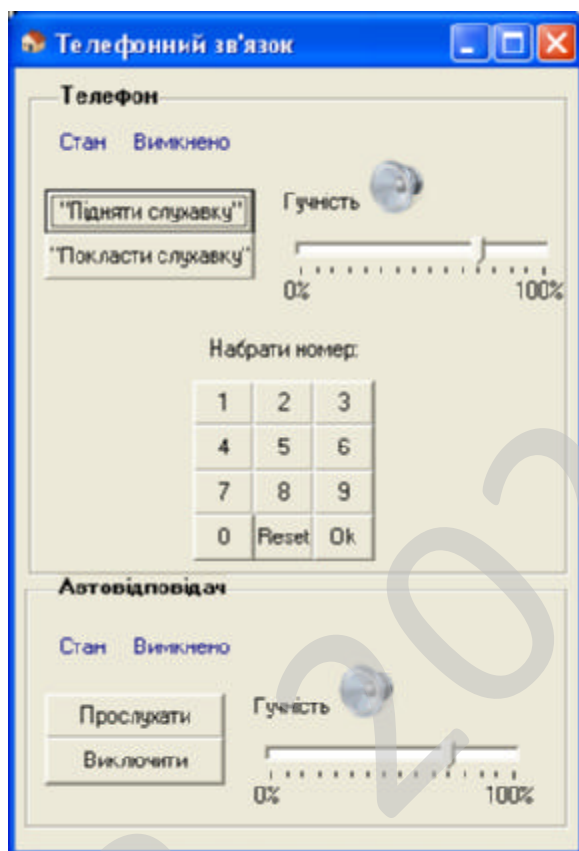


Рисунок 5.7 – Управління телефонним зв'язком

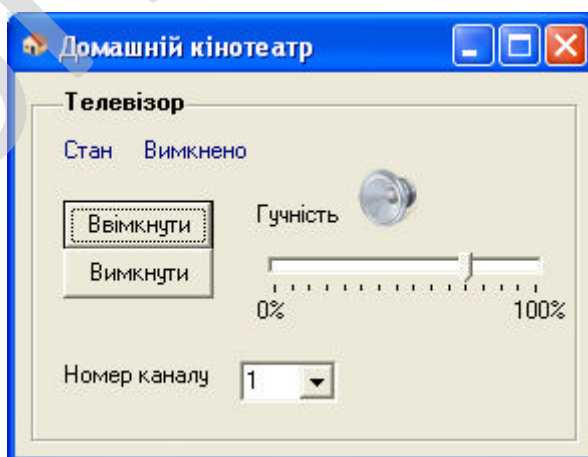


Рисунок 5.8 – Управління домашнім кінотеатром

Вікно довідки зображене на рисунку 5.9.

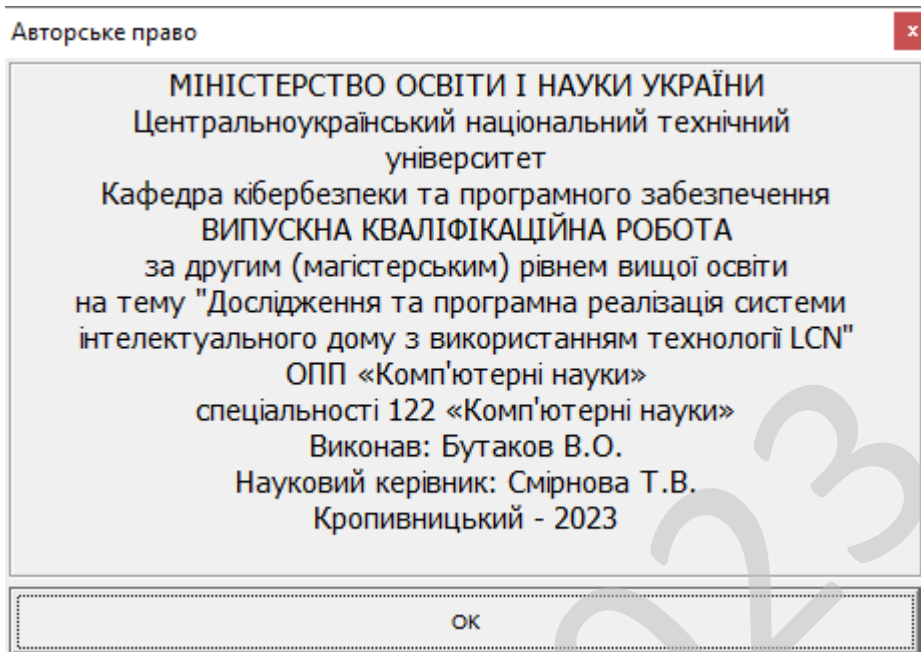


Рисунок 5.9 – Довідка про програму

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального дому з використанням технології LCN.

Метою розробки є дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN.

Об'єктом дослідження є процес інтелектуального дому з використанням технології LCN.

Предметом дослідження є методи інтелектуального дому з використанням технології LCN.

Методи дослідження базуються на методах Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуального дому з використанням технології LCN.

– Розроблено вітчизняний продукт інтелектуального дому з використанням технології LCN, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи інтелектуального дому з використанням технології LCN.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	280
3. Запланований термін розробки, днів	Frq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	28000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	64	Ф 7.1-7.4
Впровадження	13	Д13
Всього	105	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз} \cdot N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{105 \cdot 1}{60 - 5} = 1,9 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{\text{др}}^c = \frac{3_{\text{ч}} \cdot n_{\text{міс}}}{1,2}, \quad (7.6)$$

$$\Phi_{\text{др}}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{nv} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{nv} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Компбест за 06.11.23 – джерело <https://compbest.com.ua/>.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	INTEL Core™ i3 10105 (BX8070110105) 1200, 4 ядра, 8 потоків, 3.7 GHz, 4.4 GHz, TDP - 65 Вт, 14nm, BOX	-
Системна плата	ASUS PRIME H510M-K сокет - 1200, DDR4, 64 ГБ, 3200 MHz, LAN - 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Intel UHD Graphics 630	-
Жорсткий диск	SSD M.2 2280 240GB Apacer (AP240GAS2280P4-1) 240 GB, 3D TLC, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Fury Beast Black Kingston Fury (KF432C16BB/8)	-
Блок живлення	Gamemax 500W (GM-500B) ATX 12V v2.3, 500 Вт, 20+4 pin, CPU - 4+4pin, GPU - 1x6 pin, SATA - 3, Peripheral - 2, +12V1 - 20A, 1x120 мм, 150 x 140 x 86 мм	-
Корпус	Vinga CS210B, Miditower, ATX, Micro - ATX, Mini - ITX	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	—	—
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	199177

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$K_p = 1760367$		$A_p = 190286$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 800 \cdot 105 / 280 = 300 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 300 \cdot 10 \cdot 0,01 = 30 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(300 + 30) = 73 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджів, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Згідно виданих викладачем норм приймаємо пів пачки паперу на три місяці розробки ($n_p=0,33$). Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 210$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$З_{M1} = Ц_n \cdot n_p \cdot N_m. \quad (7.16)$$

$$З_{M1} = 210 \cdot 0,5 = 105 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 100 екземплярів програм:

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 33,6 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 44 грн./шт.

$$З_{M2} = 33,6 \cdot 100 = 3360 \text{ грн.}$$

Згідно виданих норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{M3} = \sum Ц_{з}, \quad (7.18)$$

де: $Ц_{з}$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$З_M = (105 + 3360 + 1702) / 280 = 18 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = З_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 280$ прим.):

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 190286 \cdot 3 / (280 \cdot 12) = 170 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 300 + 30 + 73 + 45 + 18 + 45 + 170 = 681 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	Z_o	300
2. Додаткова зарплата виконавців	Z_d	30
3. Відрахування на соціальні потреби	C_{oc}	73
4. Загальногосподарські витрати	Γ_{ocn}	45
5. Витрати на матеріали	Z_m	18
6. Освоєння нових операційних систем, мов програмування	O_n	45
7. Амортизація основних фондів	A_m	170
8. Повна собівартість програмного забезпечення	C_n	681
9. Плановий прибуток	P_p	341
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1022
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	204,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1226,4

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 681 = 341 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1226
Всього капітальних витрат	–	1226

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	15190	4500
2. Витрати на електроенергію	$Z_{ел}$	2692	1864
3. Витрати на амортизацію	$Z_{ам}$	0	307
Всього витрат за рік	I	17882	6671

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення витрати на обслуговування системи зменшились з 15190 грн до 4500 грн на рік.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	1226	–	306,5
Всього відрахувань	-	–	1226	–	306,5

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел} \quad (7.24)$$

$$Z_{ел баз} = 0,545 \cdot 1300 \cdot 3,8 = 2692 \text{ грн.}$$

$$Z_{ел нов} = 0,545 \cdot 900 \cdot 3,8 = 1864 \text{ грн.}$$

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (Ц_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (1022 - 681) \cdot 280 - (0,05 \cdot 1408000 + 0,5 \cdot 199177 + 0,25 \cdot 33190 + 0,1 \cdot 28000) \cdot 3/12 = 50208 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(Ц_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p – балансова вартість основних фондів розробника.

$$T_e = \frac{1760367}{(1022 - 681) \cdot 280 \cdot 12 / 3} = 4,6 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n (K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

$$E_{cn} = (17882-6671) \cdot 0,25 \cdot 1226 = 10904 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	280
2. Повна собівартість розробленої програми	Грн.	681
3. Ціна розробленої програми	Грн.	1022
4. Плановий прибуток від реалізації розробленої програми	Грн.	341
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	95480
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	50208
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Роки	4,6
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	10904
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,11

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_б}{I_б - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{1226}{17882 - 6671} = 0,11 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

КБПЗ - 2023

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це система правових, соціально-економічних, санітарно-гігієнічних, лікувально-профілактичних, організаційно-технічних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини під час трудової діяльності.

З широким впровадженням автоматизації та комп'ютеризації виникла потреба врахування психологічних можливостей людини, таких як швидкість реакції, особливості пам'яті та уваги, емоційний стан та ін. Створення і розвиток операторської діяльності спричинив у фаховій структурі праці ряд суттєвих змін. Наприклад, фізична важкість праці суттєво зменшилась, як і ризик травматизму на виробництві, однак разом з тим, на працюючу людину посилюється вплив нових, раніше не відомих чи мало вивчених різних несприятливих виробничих та психоемоційних чинників, серед яких особливо важливими є фактори психофізіологічні фактори.

Охорона праці та здоров'я у сфері ІТ – це комплекс заходів, які спрямовані на забезпечення безпечних і здорових умов праці для працівників, які використовують інформаційні технології, а також на запобігання травматизму, професійним захворюванням і стресу.

Охорона праці та здоров'я у сфері ІТ включає такі напрями, як:

– Ергономіка – це наука про адаптацію робочого середовища до фізичних і психологічних особливостей людини². Ергономіка вимагає врахування таких факторів, як розміри, форми, кольори, освітлення, шум, температура, вологість, вентиляція, постава, рухи, пози, втома, навантаження на очі та ін. Ергономіка допомагає покращити комфорт, продуктивність і задоволення працівників.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

– Комп'ютерна безпека – це захист комп'ютерних систем і даних від несанкціонованого доступу, зміни, знищення або блокування. Комп'ютерна безпека вимагає використання антивірусних програм, фаєрволів, паролей, шифрування, резервного копіювання та інших технологій. Комп'ютерна безпека допомагає запобігти крадіжці, шпигунству, шантажу, саботажу та іншим загрозам.

– Соціальна взаємодія – це процес спілкування між людьми у робочому колективі або через мережеві сервіси. Соціальна взаємодія вимагає дотримання правил етикету, поваги, толерантності, співробітництва та конструктивного діалогу. Соціальна взаємодія допомагає покращити настрій, мотивацію, комунікацію та творчий потенціал працівників.

Правила охорони праці і здоров'я для програмістів:

- Регулярно роби перерви в роботі. Вставай із-за столу і розминай м'язи.
- Налаштуй яскравість і контрастність монітору так, щоб не напружувати очі.
- Використовуй ергономічну мишку і клавіатуру, які зручно лягають у руку і не викликають болю.

- Слідкуй за своєю поставою. Сиди прямо і не нахиляйся до екрану.
- Захищай свій комп'ютер від вірусів, шпигунських програм і хакерів. Оновлюй антивірусне програмне забезпечення і не відкривай підозрілі файли і посилання.

- Не забувай про соціальну взаємодію. Спілкуйся з колегами, друзями і родиною. – Відвідуй заходи, які тебе цікавлять. Не ізолюй себе від світу.
- Люби свою професію, але не забувай про інші сфери життя. Розвивай свої захоплення, хоббі і таланти. Знаходь рівновагу між роботою і відпочинком.

Закон України “Про охорону праці” визначає основні принципи, завдання, права і обов'язки суб'єктів відносин з охорони праці, а також організаційні та правові основи державного управління і контролю за дотриманням законодавства про охорону праці.

						ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			93

Згідно з цим законом, ІТ компанії повинні впроваджувати такі заходи з охорони праці:

- Створювати на підприємстві службу охорони праці або призначати відповідальних осіб, які забезпечують розроблення, реалізацію та контроль за дотриманням заходів з охорони праці.

- Забезпечувати безпечні і нешкідливі умови праці для працівників, використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби колективного та індивідуального захисту, оптимальні режими праці та відпочинку.

- Проводити атестацію робочих місць на відповідність нормативно-правовим актам з охорони праці та аудит з охорони праці.

- Проводити навчання та інструктаж з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки у разі виникнення аварії⁵.

- Забезпечувати лікувально-профілактичне обслуговування працюючих, санітарно-побутове обслуговування, пільги і компенсації для працівників, які працюють у важких і шкідливих умовах.

- Нести відповідальність за порушення законодавства про охорону праці та заподіяння шкоди життю і здоров'ю працівників.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом [1]. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- монотонність праці;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Довжина	8,56
Ширина	5,43
Висота	2.9

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6,64
Обсяг, V	м ³	не менше 20.0	19,25

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють 7 людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа приміщення у розрахунку на одного робочого місця програміста відповідає нормативним вимогам як Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», так і СанПіН 3.3.2 – 007 – 98.

А об'єм приміщення у розрахунку на одного робочого місця програміста відповідає нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», але дещо нижчий вимог СанПіН 3.3.2 – 007 – 98 (19,25 м³ замість 20 м³). Це цілком прийнятно, враховуючи хронологію та пріоритет опубліковання вищезгаданих нормативних актів.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному

приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Ia, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °C	Воло- гість,%	Швидкість повітря, м/с	Температура, °C	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	55-65	0,1

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосується у холодний період року.

В літню пору застосується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

– бажано, щоб робочий стіл при необхідності можна було регулювати по висоті в межах 680-780 мм, а висота над рівнем підлоги робочої поверхні, на якій працює програміст, повинна складати 720 мм.

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Додаткові рекомендовані заходи: забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення та занулення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

8.5 Розрахунок занулення

Розрахунок занулення.

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність;
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус;
3. Розрахунок робочого і повторного заземлювачів

Початкові дані

1. Потужність електроприладів, які підлягають зануленню:

$P = 8$ кВт.

2. Довжина магістрального кабеля: $L_M = 32$ м.

3. Довжина розгалуження (від розподільного щита до електроприладів) :
 $l = 15$ м.

4. Матеріал провідників кабеля—алюміній.

5. Лінійна напруга $U = 380$ В.

6. Фазна напруга $U_f = 220$ В.

						ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			99

2. Охорона праці. Ч. 2. Занулення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM–сумісного типу / [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака, А. Е. Солових, С. Е. Катеринич] ; Мін-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – 2–ге вид., перероб. та доп. – Кропивницький : ЦНТУ, 2019. – 27 с. URI: <http://dspace.kntu.kr.ua/jspui/handle/123456789/8769>

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.23).

4. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.23).

5. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0093-98#Text> (дата звернення 19.09.23).

6. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.09.23).

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуального дому з використанням технології LCN.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтелектуального дому з використанням технології LCN.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуального дому з використанням технології LCN.

– Досліджена система інтелектуального дому з використанням технології LCN.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального дому з використанням технології LCN.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального дому з використанням технології LCN.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

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.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

При створенні 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. Це 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 ДСТУ 8845:2019.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 10904 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,11 роки.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бутаков В.О. Дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.

2. Peter Hoddie, Lizzie Prader «IoT Development for ESP32 and ESP8266 with JavaScript: A Practical Guide to XS and the Moddable SDK» ISBN-13 (pbk): 978-1-4842-5069-3 ISBN-13 (electronic): 978-1-4842-5070-9

3. STM32CubeMX for STM32 configuration and initialization C code generation. User manual. June 2022. 397 p.

4. І.В.Чихіра, А.Г. Микитишин Конспект лекцій з дисципліни «Програмування систем реального часу» / Укладачі : Чихіра І.В., Микитишин А.Г., – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя , 2016. – 76 с.

5. Jack Ganssle and Michael Barr. 2003. Embedded Systems Dictionary. CMP Books.

6. 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.

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

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

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

10. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

11. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

12. 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.

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

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. Smirnov O. Kuznetsov A., Zaichenko 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.

21. 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.

22. 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.

23. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. Smirnov A.A., Kuznetsov 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

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

36. 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

37. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.

38. 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.

39. Смірнов О.А., Коноплицька-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.

40. 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.

41. Смірнов О.А., Дреєва Г.М., Дреєв 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.

42. Смірнов О.А., Коноплицька-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.

43. 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).

44. Смірнов О.А., Дреєва 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

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

45. Дреєва Г.М., Смірнов О.А., Дреєв 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.

46. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв 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.

47. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-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.

48. 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.

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

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

51. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

					ВКРМ-122.23.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-122.23.0003.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Бутаков В.О.				Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.						
Н. Контр.	Коваленко А.С.				М	1	6
Затв.	Смірнов О.А.				ЦНТУ КН-22М-1		
					Дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN		

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інтелектуального дому з використанням технології LCN.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 32-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інтелектуального дому з використанням технології LCN.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) 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;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.23.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи інтелектуального дому з використанням технології LCN;
- 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.

					ВКРМ-122.23.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

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 архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

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.

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++.

					ВКРМ-122.23.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-122.23.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 110 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 19.12.2023 р.

					ВКРМ-122.23.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи інтелектуального дому з використанням технології LCN*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 48

Літера: РП

Кропивницький – 2023 року

Основна програма

Файл Project_IntellectHome.cpp основної програми

```

#include <vcl.h>
#pragma hdrstop
//-----
USEFORM("main.cpp", Form_main); // Головне вікно
USEFORM("about.cpp", Form_about); // Вікно даних про розробника
USEFORM("light.cpp", Form_light); // Вікно управління світлом
USEFORM("water.cpp", Form_water); // Вікно управління водою
USEFORM("gas.cpp", Form_gas); // Вікно управління газом
USEFORM("security.cpp", Form_security); // Вікно управління захистними системами
USEFORM("climate.cpp", Form_climate); // Вікно управління кліматом
USEFORM("phone.cpp", Form_phone); // Вікно управління телефонією
USEFORM("tv.cpp", Form_tv); // Вікно управління телевізорами
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        // Створення головної форми
        Application->CreateForm(__classid(TForm_main), &Form_main);
        // Створення форми даних про розробника
        Application->CreateForm(__classid(TForm_about), &Form_about);
        // Створення форми управління світлом
        Application->CreateForm(__classid(TForm_light), &Form_light);
        // Створення форми управління водою
        Application->CreateForm(__classid(TForm_water), &Form_water);
        // Створення форми управління газом
        Application->CreateForm(__classid(TForm_gas), &Form_gas);
        // Створення форми управління захистними системами
        Application->CreateForm(__classid(TForm_security),
&Form_security);
        // Створення форми управління кліматом
        Application->CreateForm(__classid(TForm_climate),
&Form_climate);
        // Створення форми управління телефонією
        Application->CreateForm(__classid(TForm_phone), &Form_phone);
        // Створення форми управління телевізорами
        Application->CreateForm(__classid(TForm_tv), &Form_tv);
        // Створення форми запуску проекту
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0;
}
//-----

```

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 "Управління 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"
```

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

КБПЗ - 2023

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:        // Компоненти IDE-управління  
        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:        // Визначено користувачем  
public:        // Визначено користувачем  
    __fastcall TForm_main(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm_main *Form_main;  
//-----  
#endif
```

Файл light.cpp - керування освітленням

```

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

#include "light.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_Command(0,1,18,1); //0-код будинку; 1-код пристрою, що
    керує ліхтарем; 18-код команди "on"; 1-кількість повторних надсилань команди
    ClientSocket->SendLeviton_LCN_(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_LCN_Command(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_LCN_Command(0,2,18,1);
    ClientSocket->SendLeviton_LCN_(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;
}

```

```

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

ClientSocket->Send_LCN_Command(0,3,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,4,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,5,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,6,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,7,18,1);
ClientSocket->SendLeviton_LCN_(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_LCN_Command(0,8,18,1);

```

```
ClientSocket->SendLeviton_LCN_(0,8,TrackBar_bathroom->Position);
}
```

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

```
//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_LCN_Command(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_LCN_Command(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_LCN_Command(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_LCN_Command(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_LCN_Command(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_LCN_Command(0,7,19,1);
}
//-----
```

```

}
//-----

//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_LCN_Command(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_LCN_Command(0,1,18,1);
ClientSocket->SendLeviton_LCN_(0,1,TrackBar_torch->Position);

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

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

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

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

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

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

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

TrackBar_torch->Enabled=true;

```

```

TrackBar_bedroom->Enabled=true;
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_LCN_Command(0,1,19,1);

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

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

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

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

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

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

bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
ClientSocket->Send_LCN_Command(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
//ввимкнення та вимкнення світла випадковим чином
void __fastcall TForm_light::Button1Click(TObject *Sender)
{

```

```

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_LCN_Command(0,x,18,1);
y=random (6)+2;
ClientSocket->Send_LCN_Command(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_LCN_(0,1,TrackBar_torch->Position);
}
//-----

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

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

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

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

```

```

//зміна яскравості освітлення спальні
void __fastcall TForm_light::TrackBar_bedroomChange(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(0,2,TrackBar_bedroom->Position);
}
//-----

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

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

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

ClientSocket->Send_LCN_StatusQuery();

st=_LCN_State->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=_LCN_State->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=_LCN_State->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;
}
}

```

```
st=_LCN_State->GetStateOnOff(0, 4);
if(st=="On") {
kitchen->Caption="Ввимкнено";
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=_LCN_State->GetStateOnOff(0, 5);
if(st=="On") {
cabinet->Caption="Ввимкнено";
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=_LCN_State->GetStateOnOff(0, 6);
if(st=="On") {
bedroom->Caption="Ввимкнено";
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=_LCN_State->GetStateOnOff(0, 7);
if(st=="On") {
baby_room->Caption="Ввимкнено";
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=_LCN_State->GetStateOnOff(0, 8);
if(st=="On") {
bathroom->Caption="Ввимкнено";
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;
}
}
```

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:      // Компоненти IDE-управління
        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;

```

```

TLabel *Label10;
TBevel *Bevel2;
TLabel *Label11;
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: // Визначено користувачем
public: // Визначено користувачем
    __fastcall TForm_light(TComponent* Owner);
};
//-----

```

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

К6П3-2023

Файл climate.cpp - керування кліматом

```

#include <vcl.h>
#pragma hdrstop

#include "climate.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_Command(0,21,18,1);
}
//-----

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

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

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

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

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

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

```
void __fastcall TForm_climate::TrackBar5Change(TObject *Sender)
{
ClientSocket->SendLeviton_LCN_(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_LCN_StatusQuery();
st=_LCN_State->GetStateOnOff(0, 21);
if(st=="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LCN_State->GetStateOnOff(0, 22);
if(st=="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LCN_State->GetStateOnOff(0, 23);
if(st=="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";
st=_LCN_State->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=_LCN_State->GetStateOnOff(0, 26);
if(st=="On") Label_5->Caption="Відкрито"; else Label_5->Caption="Закрито";
st=_LCN_State->GetStateOnOff(0, 27);
if(st=="On") Label_6->Caption="Відкрито"; else Label_6->Caption="Закрито";
st=_LCN_State->GetStateOnOff(0, 28);
if(st=="On") Label_7->Caption="Відкрито"; else Label_7->Caption="Закрито";
st=_LCN_State->GetStateOnOff(0, 29);
if(st=="On") Label_8->Caption="Відкрито"; else Label_8->Caption="Закрито";
}
```

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:      // Компоненти IDE-управління
    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;

```

```
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;
```

```
TEdit *Edit2;
TLabel *Label185;
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: // Визначено користувачем
public: // Визначено користувачем
    __fastcall TForm_climate(TComponent* Owner);
};
//-----
extern PACKAGE TForm_climate *Form_climate;
//-----
#endif
```

Файл water.cpp - керування системою водопостачання

```

#include <vcl.h>
#pragma hdrstop

#include "water.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_Command(0,9,18,1);
}
//-----

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

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

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

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

```

```

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

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

//закриття клапану гарячої води на кухні
void __fastcall TForm_water::Button11Click(TObject *Sender)
{
ClientSocket->Send_LCN_Command(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_LCN_StatusQuery();
st=_LCN_State->GetStateOnOff(0, 9);
if(st=="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LCN_State->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=_LCN_State->GetStateOnOff(0, 10);
if(st=="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";
st=_LCN_State->GetStateOnOff(0, 11);
if(st=="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";

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

```

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:      // Компоненти IDE-управління
    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:      // Визначено користувачем
public:      // Визначено користувачем
    __fastcall TForm_water(TComponent* Owner);
};

```

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

К6П3 - 2023

Файл gas.cpp - керування системою газопостачання

```

#include <vcl.h>
#pragma hdrstop

#include "gas.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_Command(0,15,18,1);
}
//-----

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

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

void __fastcall TForm_gas::Button3Click(TObject *Sender)
{
ClientSocket->Send_LCN_Command(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_LCN_StatusQuery();
st=_LCN_State->GetStateOnOff(0, 15);
if(st=="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";
st=_LCN_State->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";

}

```

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:      // Компоненти IDE-управління
    TGroupBox *GroupBox1;
    TLabel *Label1;
    TLabel *Label_1;
    TLabel *Label13;
    TLabel *Label_11;
    TButton *Button1;
    TButton *Button2;
    TGroupBox *GroupBox2;
    TLabel *Label3;
    TLabel *Label_2;
    TLabel *Label15;
    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:         // Визначено користувачем
public:          // Визначено користувачем
    __fastcall TForm_gas(TComponent* Owner);
};
//-----
extern PACKAGE TForm_gas *Form_gas;
//-----
#endif

```

Файл tv.cpp - керування домашнім кінотеатром

```

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

#include "tv.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_(0,32,TrackBar1->Position);
}
//-----

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

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

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

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

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

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

}
//-----

```

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:      // Компоненти IDE-управління
    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:         // Визначено користувачем
public:         // Визначено користувачем
    __fastcall TForm_tv(TComponent* Owner);
};
//-----
extern PACKAGE TForm_tv *Form_tv;
//-----
#endif

```

Файл phone.cpp - керування телефонним зв'язком

```

#include <vcl.h>
#pragma hdrstop

#include "phone.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_(0,30,TrackBar1->Position);
}
//-----

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

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

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

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

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

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

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

```

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

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

}
//-----
```

КБПЗ - 2023

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:      // Компоненти IDE-управління
    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:      // Визначено користувачем
public:      // Визначено користувачем
    __fastcall TForm_phone(TComponent* Owner);
};
extern PACKAGE TForm_phone *Form_phone;
#endif

```

Файл security.cpp - керування системою безпеки

```

#include <vcl.h>
#pragma hdrstop

#include "security.h"
#include "CPU_LCN_Socket"
//-----
#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_LCN_Command(0,17,18,1);
}
//-----

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

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

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

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

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

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

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

//визначення та виведення стану системи безпеки

```

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

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

    st=_LCN_State->GetStateOnOff(0, 19);
    if(st="On") Label_3->Caption="Ввімкнено"; else Label_3->Caption="Вимкнено";
    st=_LCN_State->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";
}
```

КБГЗ - 2023

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:      // Компоненти IDE-управління
    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: // Визначено користувачем
public:  // Визначено користувачем
    __fastcall TForm_security(TComponent* Owner);
};
extern PACKAGE TForm_security *Form_security;
#endif

```

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("МАГІСТЕРСЬКА РОБОТА");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("на тему:");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("Дослідження та програмна реалізація системи інтелектуального
дому з використанням технології LCN");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("Керівник: Смірнова Т.В.");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("Розробив: студент Бутаков Владислав Олександрович");
    Memo1->Lines->Add("
гр. КН-22М-1");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("");
    Memo1->Lines->Add("м. Кропивницький 2023");
}
}
```

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:      // Компоненти IDE-управління  
        TImage *Image1;  
        TMemo *Memo1;  
        TButton *Button1;  
        void __fastcall Button1Click(TObject *Sender);  
private:      // Визначено користувачем  
public:      // Визначено користувачем  
    __fastcall TForm_about(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm_about *Form_about;  
//-----  
#endif
```

Файл CPU_LCN_Socket.cpp - протокол LCN

```

//-----
#include <basepch.h>

#pragma hdrstop

#include "CPU_LCN_Socket.h"

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

static inline void ValidCtrCheck(TCPU_LCN_Socket *)
{
    new TCPU_LCN_Socket(NULL);
}
//#include "OcelotStateUnit.cpp"
//-----
__fastcall TCPU_LCN_Socket::TCPU_LCN_Socket(TComponent* Owner)
    : TClientSocket(Owner)
{
    _LCN_StateChangeNum=new T_LCN_StateChangeNum;
    OcelotStateChangeNum=new TOcelotStateChangeNum;
    _LCN_State=new T_LCN_State;
    OcelotState=new TOcelotState;

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

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

__fastcall TCPU_LCN_Socket::~TCPU_LCN_Socket(void)
{
    delete _LCN_StateChangeNum;
    delete OcelotStateChangeNum;
    delete _LCN_State;
    delete OcelotState;
}
//-----
namespace CPU_LCN_Socket
{
    void __fastcall PACKAGE Register()
    {
        TComponentClass classes[1] = {__classid(TCPU_LCN_Socket)};
        RegisterComponents(" CPU-XA", classes, 0);
    }
}
//-----
void __fastcall TCPU_LCN_Socket::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);
}

```

```

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

//-----
void __fastcall TCPU_LCN_Socket::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_LCN_Socket::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)
        {

```

```

        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(&_LCN_StateChangeNum->Num,&local_pointer[1],2);
        _LCN_State->LoadInfoBuff(&local_pointer[3]);
        if (FOnReadl0StatusData)
            FOnReadl0StatusData(this,true);
    }
    if (CommandStrSize==3)
        if (FOnReadl0StatusData)
            FOnReadl0StatusData(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_LCN_Socket::SendBuf(char *buf,long size)
{
    if (Active)
    {
        char *buff;
        long size_buff=size+7;
        buff=(char *)malloc(size_buff);

```

```

    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_LCN_Socket::SendOcelotStatusQuery(void)
{
    char buf[3];
    buf[0]='q';
    memcpy(&buf[1], &OcelotStateChangeNum->Num, 2);
    SendBuf(buf, 3);
}
//-----
void __fastcall TCPU_LCN_Socket::Sendl0StatusQuery(void)
{
    char buf[3];
    buf[0]='x';
    memcpy(&buf[1], &_LCN_StateChangeNum->Num, 2);
    SendBuf(buf, 3);
}
//-----
unsigned short __fastcall TCPU_LCN_Socket::GetDataFromUnit(unsigned char
UnitNumber)
{
    return OcelotState->GetDataFromUnit(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_LCN_Socket::GetIO(unsigned char
UnitNumber, unsigned char Point)
{
    return OcelotState->GetIO(UnitNumber, Point);
}
//-----
unsigned char __fastcall TCPU_LCN_Socket::GetUnitVer(unsigned char UnitNumber)
{
    return OcelotState->GetUnitVer(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_LCN_Socket::GetUnitType(unsigned char UnitNumber)
{
    return OcelotState->GetUnitType(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_LCN_Socket::GetUnitsCount(void)
{
    return OcelotState->GetUnitsCount();
}
//-----
TDateTime __fastcall TCPU_LCN_Socket::GetCPUXADateTime()
{
    return OcelotState->GetCPUXADateTime();
}
//-----
unsigned short __fastcall TCPU_LCN_Socket::GetVariable(unsigned char VarNumber)
{
    return OcelotState->GetVariable(VarNumber);
}
//-----
unsigned short __fastcall TCPU_LCN_Socket::GetTimer(unsigned char TimerNumber)
{
    return OcelotState->GetTimer(TimerNumber);
}
//-----

```

```

TDateTime TCPU_LCN_Socket::GetTimerAsTime(unsigned char VarNumber)
{
    return UShortToTime(OcelotState->GetTimer(VarNumber));
}
//-----
TDateTime TCPU_LCN_Socket::GetVarAsTime(unsigned char VarNumber)
{
    return UShortToTime(OcelotState->GetVariable(VarNumber));
}
//-----
TDateTime TCPU_LCN_Socket::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_LCN_Socket::SetTimeAsVar(unsigned char VarNumber, TDateTime time)
{
    SetVariable(VarNumber,TimeToUShort(time));
}
//-----
void TCPU_LCN_Socket::SetTimeAsTimer(unsigned char VarNumber, TDateTime time)
{
    SetTimer(VarNumber,TimeToUShort(time));
}
//-----
unsigned short TCPU_LCN_Socket::TimeToUShort(TDateTime time)
{
    Word Hour, Min, Sec, MSec;
    DecodeTime(time, Hour, Min, Sec, MSec);
    return Hour*100+Min;
}
//-----
void __fastcall TCPU_LCN_Socket::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_LCN_Socket:: 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);
}
//-----

```

```

void __fastcall TCPU_LCN_Socket::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_LCN_Socket::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_LCN_Socket::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_LCN_Socket::SendIr(unsigned short Value)
{
char buf[4];
buf[0]='c';
buf[1]=5;
memcpy(&buf[2],&Value,2);
SendBuf(buf,4);
}
//-----
void __fastcall TCPU_LCN_Socket::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_LCN_Socket::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_LCN_Socket::GetInternalProtocolVersion(void)

```

```

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

void __fastcall T_LCN_StateChangeNum::Next(void)
{
if (Num==65534)
    Num=1;
else
    Num++;
};
//-----

__fastcall T_LCN_StateChangeNum::T_LCN_StateChangeNum(void)
{
Num=1;
};
//-----

AnsiString __fastcall T_LCN_State::GetStateOnOff(unsigned char house, unsigned
char key)
{
if (map[house][key]==true)
return "On";
else
return " --";
}
//-----

bool __fastcall T_LCN_State::GetOnOffStatus(unsigned char house, unsigned char
key)
{
if (map[house][key]==true)
return true;
else
return false;
}
//-----

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

void __fastcall TCPU_LCN_Socket::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_LCN_Socket::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;
}

```

Файл CPU_LCN_Socket.h - бібліотека для файлу CPU_LCN_Socket.cpp

```

//-----
#ifndef CPU_LCN_Socket
#define CPU_LCN_Socket
//-----
#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 *TOnReadl0StatusData) (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_LCN_StateChangeNum {
public:          // Визначено користувачем
unsigned short Num;
void __fastcall Next(void);
__fastcall T_LCN_StateChangeNum(void);
};

class T_LCN_State {
private:
unsigned char ActiveKey[16];
public:          // Визначено користувачем
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_LCN_Socket : public TClientSocket
{
private:
    AnsiString FVersion;
    AnsiString FLogin;
    AnsiString FPassword;
    unsigned char FAuth;
    TOcelotStateChangeNum *OcelotStateChangeNum;
    T_LCN_StateChangeNum *_LCN_StateChangeNum;
    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;
    }
}

```

```

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_LCN_Socket(TComponent* Owner);
__fastcall ~TCPU_LCN_Socket(void);
T_LCN_State *_LCN_State;
// запит на одержання даних про статуси
void __fastcall SendOcelotStatusQuery(void);
void __fastcall Send10StatusQuery(void);
// Перенос методів 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);
// Відправка команд
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};

```

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

K6П3-2023