

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
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 керування
компонентами інтелектуального будинку за технологією
TELETASK”

Виконав здобувач вищої освіти
II курсу, групи КН-22МЗ
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Разумовський М.Д.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
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

Разумовському Максиму Дмитровичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Лисенко Ірина Анатоліївна, канд. техн. наук</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 37-13 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 керування компонентами інтелектуального будинку за технологією TELETASK</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

Разумовський М.Д. Дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK. 122 Комп'ютерні науки. Центральнотраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування компонентами інтелектуального будинку за технологією TELETASK.

Метою розробки є дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

Об'єктом дослідження є процес керування компонентами інтелектуального будинку за технологією TELETASK.

Предметом дослідження є методи керування компонентами інтелектуального будинку за технологією TELETASK.

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

Результат роботи – програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

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 будинок, TELETASK

ABSTRACT

Razumovskyi M.D. Research and software implementation of the system for controlling the components of an intelligent house using the TELETASK 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 control system of the components of an intelligent house using the TELETASK technology.

The purpose of the development is the research and software implementation of the system for controlling the components of an intelligent house using the TELETASK technology.

The object of the study is the process of controlling the components of an intelligent house using the TELETASK technology.

The subject of the research is methods of controlling the components of an intelligent house using the TELETASK 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 system for controlling the components of an intelligent house using the TELETASK 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, intelligent house, TELETASK

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	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	17
3.1 Опис функціонування системи	17
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	44
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	44
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 НАУКОВА НОВИЗНА	63

						ВКРМ-122.23.0090.00.00.ПЗ		
Вим	Арк	№ докум.	Підп.	Дата		Літ.	Аркуш	Аркушів
Розроб.	Разумовський М.Д.				Дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK	М	1	103
Перев.	Писенко І.А.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	64
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	64
7.2 Розрахунок трудомісткості розробки програмної продукції.....	66
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	68
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	72
7.5 Визначення собівартості розробки та ціни програмної продукції.....	77
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	80
7.7 Визначення експлуатаційних витрат.....	80
7.8 Визначення економічної ефективності програмної продукції.....	82
7.9 Висновок.....	84
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	85
8.1 Вступ.....	85
8.2 Пожежна безпека.....	87
8.3 Характеристика умов праці програміста	89
8.4 Розробка заходів з умов поліпшення охорони праці.....	91
8.5 Розрахункова частина	91
9 ОСНОВНІ ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Системи "інтелектуального будинку" або, інакше кажучи, домашньої автоматизації, сьогодні дозволяють робити дуже багато чого – від контролю світла й температури до керування звуком аудіосистем і безпекою будинку й родини, використовуючи для цього звичайні мобільні пристрої.

Технології "інтелектуального будинку" дозволяють сьогодні контролювати практично все. Деякі речі, такі як, наприклад, пристрої для включення/вимикання світла за командою, прості й недорогі, інші, наприклад, системи відеоспостереження, вимагають більше значних вкладень. Ще недавно системи Smart Home залучали винятково багатіїв, сьогодні це вже, як говорять, мейнстрим, тобто звичні багато пристроїв. І розвиток цих технологій, що одержало потужний поштовх за рахунок широкого поширення широкополосного мобільного Інтернету, іде дуже швидко, не дарма на цей ринок один за іншим пішли електронні гіганти, такі як Apple, Google і Samsung.

Природно, складні й об'ємні системи, які контролюють усе в будинку й довкола нього, до якого ви одержуєте доступ через смартфон або планшет, є в продажі й, якщо у вас досить грошей, ви цілком можете їх установити в себе. При цьому сьогодні практично кожна компанія, що займається домашніми пристроями й системами, або системами безпеки, хоче відкусити шматочок пирога ринку домашньої автоматизації й починає випускати дистанційно керовані продукти. Але поки, на жаль, немає систем, які могли б об'єднати всі пристрої, поза залежністю хто їх робить і за яким принципом вони працюють.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Огляд існуючих систем керування компонентами інтелектуального будинку за технологією TELETASK.

– Дослідження системи керування компонентами інтелектуального будинку за технологією TELETASK.

– Програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

Об'єктом дослідження є процес керування компонентами інтелектуального будинку за технологією TELETASK.

Предметом дослідження є методи керування компонентами інтелектуального будинку за технологією TELETASK.

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 керування компонентами інтелектуального будинку за технологією TELETASK.

– Розроблено вітчизняний продукт керування компонентами інтелектуального будинку за технологією TELETASK, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування компонентами інтелектуального будинку за технологією TELETASK.

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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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 керування компонентами інтелектуального будинку за технологією TELETASK, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					VKPM-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для керування компонентами інтелектуального будинку за технологією TELETASK. Компанія TELETASK (Бельгія) – виробник устаткування для Систем Домашньої Автоматизації (СДА).

Головний офіс компанії й виробництво перебуває в Бельгії. Офіс розташований у місті Гент. Компанія має представництва й партнерів більш ніж в 40 країнах, у всіх регіонах миру. Компанія, на відміну від інших фірм, що роблять устаткування для систем Інтелектуальний будинок, споконвічно займалася й розробляла системи для домашньої автоматизації – Інтелектуальний будинок. У цьому її принципова, істотна й сильна позитивна відмінність. Шлях, пройдений TELETASK, дозволяє їй робити якісне й зручне встаткування для будинків і приміщень, де людина перебуває тривалий час: квартири, заміські будинки, офіси, готелі, салони, магазини й т.д.

У світовому арсеналі компанії десятки тисяч об'єктів, обладнаних системою TELETASK. Це будівельні проекти різного профілю й розміру. Від квартир для соціальних верств населення (будинки для людей похилого віку, пансіонати) до розкішних готелів і апартаментів на різних континентах, аж до салонів дорогих автомобільних брендів і навіть аеропортів. Площею від 25 м.кв. до 35000 м.кв.

1.2 Область застосування

Областю застосування системи є інтелектуальні будинки. Автоматизація полегшує й поліпшує ваше життя в будинку або квартирі, більше того, вона дозволяє заощаджувати час і гроші. От кілька доводів на користь початку створення свого інтелектуального будинку.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Автоматизація підвищує ефективність. Це відноситься як до ефективності окремих систем у вашому будинку, так і до ефективності всього господарства. Наприклад, оскільки система автоматизації здатна управляти декількома пристроями, ви можете одним натисканням кнопки відключити термостати й виключити світло, коли вирішите поїхати у відпустку. Не треба буде хвилюватися, що ви щось забули зробити.

Автоматизація заощаджує гроші. Тільки інтелектуальне керування освітленням, опаленням, вентиляцією або кондиціонуванням дозволяє знизити витрати на електроенергію на 15-20%, за твердженнями розроблювачів систем автоматизації. Для власників власних будинків сюди додається й можливість використання нестандартних систем обігріву або охолодження, що також різко зменшує витрати на електрику й дозволяє швидко окупити інвестиції в "інтелектуальний будинок".

Автоматизація зручна. Можливість контролювати безліч електронних пристроїв (світло, опалення, аудіо/відео, штори й двері, замки, системи безпеки й т.д.) за допомогою одного сенсорного пристрою на стіні або просто смартфона – уже одне це змушує задуматися про автоматизацію будинку.

Автоматизація створює комфорт. Можливість активно контролювати різні електронні елементи будинку забезпечує комфорт, при якому навколишня вас середовище – від установки освітлення до включення музичного центра – починає вам підкорятися.

Автоматизація забезпечує спокій. Система контролю домашніх пристроїв дозволяє уникнути проблем, які можуть виникнути по безпам'ятності або через невміння. Ви з будь-якого місця можете перевірити, чи всі нормально у вашім будинку й, при необхідності, провести корекцію за допомогою смартфона.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

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

Огляд тенденцій у сфері технологій "інтелектуального будинку", які будуть переважати в 2024 році

Ідея "інтелектуального будинку" продовжує еволюціонувати. Всі нові стартапи представляють свої інноваційні продукти, продукти, що одержали у свій час фінансування за допомогою краудфандингових проєктів, почали серійно вироблятися, великі компанії одна за однією приєднуються до гравців цього ринку. Все це, а також технології, представлені зовсім недавно на виставці CES 2024, дозволяють нам зрозуміти тенденції, які ми будемо спостерігати в сфері систем "інтелектуального будинку" у поточному, 2024 року.

Краудфандинг став мейнстримом і буде усе сильніше впливати на ринок

Краудфандинг або народне фінансування проєктів розробки й виробництва різних систем і пристроїв розбурхав більш-менш спокійний і передбачуваний мир інвестиційних організацій. Тільки за минулий рік краудфандинг додав більше 65 млрд. доларів у світову економіку (мається на увазі всі проєкти, що не тільки мають відношення до високих технологій) і цей молодий ринок починає робити все більший вплив на розвиток сфери домашньої автоматизації й Інтернету речей.

Провідні платформи народного фінансування, такі як Kickstarter або Indiegogo, дозволяють дрібним стартапам або навіть окремим розроблювачам реалізувати свої ідеї, зробити продукт, що надалі може вплинути на розвиток

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

технологій, на виникнення нових інноваційних компаній, на впровадження технологій у наше повсякденне життя.

Цього року варто очікувати подальшого росту популярності краудфінансingu (навіть ми піддалися тренду й беремо участь у цьому процесі), збільшення обсягів фінансування таких проектів, посилення спеціалізації в цій сфері.

Крім того, варто очікувати зміни моделі народного фінансування, що сьогодні припускає одноразове вкладення засобів у який-небудь проект. Це добре працює на етапі створення прототипу й демонстрації його на ринку з метою перепродажу більшим і традиційним інвесторам. Але ця модель працює не для всіх. Вона не враховує великі витрати на організацію продажів і промоушн, і, як наслідок, не всі проекти можуть у підсумку вийти на фінішну пряму. І нові краудфінансінгові ресурси почнуть використовувати модель підтримки проектів, компаній, а не одиночних продуктів.

Медичні мобільні технології будуть зближатися з технологіями "інтелектуального будинку"

Сьогодні mHealth – це два великих напрямки:

- технології, пристрої й додатки для лікування й догляду за пацієнтами;
- пристрої, призначені для контролю за дотриманням здорового способу життя й фітнеса.

Обоє ці напрямки розвиваються дуже швидко, поступово зближуючись із технологіями "інтелектуального будинку" у рамках єдиної концепції Інтернету речей (Internet of Things). З'являються інтелектуальний годинник і фітнес-трекери, які можуть включатися в єдину систему домашньої автоматизації й управляти її приладами. Системи моніторингу життєвих показників працюють у єдиній мережі із системами "інтелектуального будинку", більше того, окремі медичні пристрої можуть у певних випадках запускати процеси, що ставляться до автоматизації будинку. Наприклад, система, що стежить за літніми батьками в них будинку, може у випадку падіння когось з батьків, через хаб

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

"інтелектуального будинку" відправляти повідомлення про це родичам або сусідам, включати сигналізацію або відправляти інформацію лікареві. Системи камер і датчиків, крім своєї основної охоронної функції, можуть також використовуватися для контролю активності літніх батьків і застосовуватися разом із тривожною кнопкою для літніх.

Наприклад, Evermind являє собою набір датчиків, що дозволяє родичам літньої людини непомітно й ненав'язливо відслідковувати його життєдіяльність. Датчик Evermind більше схожий на пристрій, які ми часто бачимо в комплектах автоматизації будинку – "інтелектуальну електричну вилку", тобто зовнішній модуль, що забезпечує наскрізне підключення електроустаткування до мережі. Інакше кажучи, сенсор вставляється в розетку, і при цьому він сам є розеткою. Всі дії старих контролюються й пересилаються родичам, при цьому немає нав'язливого спостереження за батьками, що вони так не люблять.

Bluetooth почне широко використовуватися, у першу чергу, у системах освітлення

В 2024 році на ринок почнуть надходити пристрої, що дозволяють використовувати Bluetooth для контролю освітлення, інтелектуальних електричних розеток і тому подібного. Ці продукти використовують технологію стільникових мереж і дозволяють підключати нові інтелектуальні пристрої також легко, як приклеїти фоторамку на стіну за допомогою липкої стрічки. Гаджети від компаній Avi-on, Oort, Seed можуть повністю змінити спосіб, що ви використовуєте для керування освітленням у будинку. Фірма Zuli почала випуск керованих через Bluetooth розеток, навіть нова версія електронного замка August може управлятися за допомогою цього протоколу.

Незважаючи на існування потужних конкурентів в особі діючих, а також розроблювальних (Thread, AllJoyn) протоколів, перспективи в Bluetooth, на нашу думку, непогані. Особливо після того, як у минулому грудні була випущена нова версія стандарту Bluetooth Smart. Серед нових функцій відзначимо підтримку IPv6, можливість використання стільникових мереж, поліпшений цикл "сон-пильнування", що дозволяє пристроям підключатися друг до друга без втручання

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

користувача. Плюс дуже зручні можливості взаємодії з пристроями різного роду, що носяться.

Керування "інтелектуальним будинком" буде здійснюватися голосом

Оскільки сьогодні майже всі просунуті смартфони оснащені системою обробки природного голосу, багато компаній, що працюють на ринку домашньої автоматизації, починають використовувати цю функцію для керування будинком за допомогою голосових команд. Це й великі компанії, такі як Nest і Apple, і невеликі – Nucleus або Ubi. Наприклад, Nest буде інтегруватися з технологією розпізнавання голосу Google Now, а "яблучний" HomeKit, природно, використовувати Siri. А інтерком Nucleus, недавно представлений однойменним стартапом, використовує голосову функцію не тільки для спілкування між людьми в будинку, але й для керування освітленням. Аналогічну функціональність вбудовує у свій додаток і компанія Ubi.

Малоспоживаючий Wi-Fi з'явиться в пристроях, навіть якщо стандарт не буде ще затверджений

Недавно дві компанії – HomeBoo і Roost – представили громадськості два різних пристрої, які використовують достоїнства малоспоживаючого Wi-Fi. Переваги його застосування цілком очевидні – вам не потрібно використовувати окремий хаб для керування таким пристроєм і працювати ці гаджети можуть практично скрізь.

Roost – це доповнення до традиційного датчика диму, що виглядає як звичайна 9-вольтова батарейка. При цьому енергопостачання – це основна, але аж ніяк не єдине завдання Roost: незвичайний гаджет покликане відправляти своєму власникові повідомлення у випадку спрацьовування датчика. Для зв'язку з роутером новинка використовує модуль Wi-Fi нового покоління, що активується лише в екстрених випадках. Подібний підхід забезпечує довгий термін служби батарейки: стандартному детектору її вистачає приблизно на п'ять років.

А HomeBoo – бездротова охоронна камера, що використовує для передачі даних також малоспоживаючий Wi-Fi. У ролі джерела живлення виступає акумулятор, одного заряду якого вистачає на 2-3 місяці роботи. Гаджет

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

виконаний у вигляді невеликої білої кулі, що кріпиться до металевої підстави за допомогою магніту.

Обидві компанії розробили свій власний варіант протоколу малоспоживаючого Wi-Fi, вирішивши не чекати появи стандарту 802.11ah.

Розширення використання систем автоматизації будинку без хаба

Ми вже обговорювали необхідність використання хаба в системі "інтелектуального будинку", а також згадали ледве вище гаджети, що працюють із використанням удосконаленого Wi-Fi і не нужденних у хабі. Ці пристрої, які в 2014 році були "гарячим продуктом", відрізняються, як правило, підтримкою декількох протоколів бездротового зв'язку й дозволяють користувачам контролювати безліч підключеним до них "інтелектуальних" систем і приладів. В 2024 році, схоже, їхній час закінчується. Навіть глава SmartThings Алекс Хоукінсон говорив про свою готовність створювати програмне забезпечення, що буде незалежним від власного хаба компанії, хоча підкреслив, що для цього треба ще якийсь час і навряд чи воно буде працювати з усіма пристроями. Уже існують аналогічні системи – наприклад, додаток Reach на базі Android, що дозволяє користувачам управляти відео, програвати музику через колонки Sonos і контролювати ряд інших пристроїв. Причому список таких пристроїв вже не малий – Philips Hue, Nest, Smart TV від Samsung, Apple і LG, Chromecast, WeMo, Dropcam, Roku, Samsung Shape і ін. Цей додаток існує поки в альфа-версії, але бета вже на підході.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Оскільки потрібно розробити просту та легку у користуванні програму, яка б виконувалась під операційною системою Windows, то для її реалізації я обрав Builder C++. Існує велике число бібліотек написаних під Builder C++ , тому це одна з важливих причин вибору мови програмування. Середовище Builder C++

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування компонентами інтелектуального будинку за технологією TELETASK.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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

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

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації 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-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Технології "інтелектуального будинку"

Перш, ніж займатися покупкою пристроїв для "інтелектуального будинку", наведемо опис технологій, які використовуються в цих пристроях. Є кілька різних стандартів або протоколів зв'язку, які використовують ці пристрої для комунікацій між собою й керуючими контролерами. Деякі пристрої з'єднуються дротовим зв'язком, деякі – бездротовою, якісь використовують і те, і інше. Обов'язково перевірте, який протокол використовує пристрій, перш ніж купувати його, щоб всі ваші покупки були сумісні один з одним:

- X10. Це самий старий із протоколів домашньої автоматизації, що з'явився ще в 70-х року минулого століття. Тепер він використовується й для дротового зв'язку й бездротовою. X10 не відрізняється безінтелектуальною швидкістю або зверхнадійним зв'язком між пристроями в мережі домашньої автоматизації, проте, цей стандарт давно налагоджений і дотепер багато хто воліють застосовувати саме X10.

- ZigBee. Це інше ім'я бездротового стандарту IEEE 802.15.4, що використовується групою виробників, що входять в ZigBee Alliance. Основним достоїнством у цьому стандарті є те, що створюється меш-мережа (стільникова мережа), у якій більшість пристроїв рівноправні й зв'язуються один з одним на рівних умовах. Стільникова мережа пропонує підвищену надійність і стійкість. Коли один з вузлів перестає функціонувати, інші вузли продовжують роботу, з'єднуючись між собою прямо або через інші проміжні вузли. Крім того, цей зв'язок споживає дуже мало електроенергії.

- Z-Wave. Ще один бездротовий протокол, що належить однієї компанії – виробникові мікросхем, у тому числі, і для систем «інтелектуального будинку» Sigma Design.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Insteon. Це, можливо, найкращий протокол, що комбінує протокол обміну даними по електромережі з бездротовим протоколом. Обоє працюють як стільникова мережа. Всі вузли мережі Insteon рівні й можуть зв'язуватися один з одним. Якщо один вузол виходить із ладу, зв'язок іде через інші. Розроблювач протоколу – компанія Smartlabs. Протокол сполучимий з X10.

- Wi-Fi. Цей мережний протокол, використовуваний тепер не тільки комп'ютерними системами. Дуже швидкий, добре працюючий. І не дивно, що частина виробників стали робити продукти для "інтелектуального будинку", що використовують достоїнства цього протоколу. Інші протоколи споживають менше енергії й використовують меншу смугу пропускання, але можливості Wi-Fi більші.

- HAI. Це протокол, використовуваний при професійних установах вартістю більше \$50 тис.

Пристрої для «інтелектуального будинку»

Статистика говорить, що найбільш популярним проектом автоматизації будинку є створення системи забезпечення безпеки. Створення домашніх кінотеатрів і систем керування музикою будинку стоїть на другому місці, за ними ідуть керування світлом, енергетикою. Потім відбувається інтеграція декількох систем.

Існує багато лінійок продуктів для різних стандартів. Тут ми покажемо тільки кілька прикладів, як нам здається, цікавих пристроїв автоматизації, які дозволять вам зробити свій будинок або квартиру "інтелектуальніше".

Nest Learning Thermostat

Термостат фірми Nest (належить Google) не тільки контролює температуру в будинку, але й прикрашає його. Його дизайн створювала людина, що розробляв iPod фірми Apple. Він працює в мережі Wi-Fi і ви можете віддаленно контролювати температуру за допомогою смартфона або комп'ютера. Він не дуже дешевий, але не буде зайвим у будь-якому «інтелектуальному будинку». Крім того, Nest має відкритий інтерфейс, що дозволить у майбутньому підключати до нього нові пристрої, як це обіцяє Google.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Бездротовий термостат Honeywell

Це інший термостат, що працює за протоколом Wi-Fi, від компанії Honeywell, що виготовляє домашні пристрої вже десятиліттями і її лінією термостатів пропонує безліч опцій підключення.

Термостат має кольоровий сенсорний екран і програмне забезпечення, здатне навчатися звичкам користувачів, а також може контролювати внутрішні й зовнішні умови для оптимізації продуктивності систем обігріву й охолодження. Налаштування термостата можуть бути задані заздалегідь за графіком, або уведені безпосередньо через сенсорний екран, комп'ютер або мобільний пристрій. На екрані відображаються кімнатна й вулична температура, значення вологості, а також місцевий прогноз погоди. Крім того, кілька версій цього пристрою можуть управлятися голосом.

Insteon

Insteon – це система автоматизації будинку, що використовує для комунікацій як електропроводку, так і бездротовий канал. Insteon комбінує двополосну й стільникову технології для створення швидкої й надійної мережі, що сумісна із чим завгодно, починаючи від термостатів і закінчуючи домашнім кінотеатром.

Набір Insteon для початківців (Insteon Starter Kit) містить у собі хаб і один модуль – диммер для регулювання світла. Хаб підключає всю систему до Інтернет (через домашній маршрутизатор) і працює з мобільним додатком, що дозволяє управляти системою звідки завгодно, програмувати таймери й власні "сценарії", а також контролювати статус системи. Хаб також може відправляти email і текстові повідомлення, коли в будинку спрацьовують датчики. За словами виробників, Insteon пропонує практично всі, від вимикачів світла й датчиків руху, до систем відкриття гаражних дверей, камер нічного бачення, датчиків витоків води й т.п.

Kevo Smart Deadbolt

Якщо ви можете відкрити свою машину без ключів, чому б вам не робити це ж з будинком або квартирою? Один з варіантів – це Kevo Smart Deadbolt

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

компанії Kwikset, ригельний замок із сенсорним кільцем зовні й інтелектуальним механізмом на базі Bluetooth 4.0 усередині. Kevo використовує відразу кілька методів для відкривання: брелок, iPhone з додатком Kevo, дотик до згаданого кільця пальцем, а також слово-пароль для відкривання.

Belkin WeMo

Це простий в установці пристрій, що може служити відмінним стартом для створення «інтелектуального будинку» і яке може почати працювати вже через кілька хвилин після інсталяції. Розетки Belkin, що працюють у мережі Wi-Fi, дозволяють дистанційно включати або виключати прилади, підключені до електричної мережі через них. Компанія Belkin випустила ще декілька дуже зручних пристроїв – це, наприклад, вимикач WeMo Home Light Switch, датчик руху, що дозволяє активувати прилади, включені в розетку WeMo, відео-нянька з доступом в Wi-Fi. Все це контролюється через мобільний додаток на базі iOS або Android.

Система освітлення Philips Hue

Hue – це система освітлення, що складає з ламп, світлових стрічок і бездротового мосту, які контролюються через додаток на базі iOS. Відповідно до інформації компанії, "кожна лампа HUE має 600 люмен світлового потоку й може робити всі варіанти білого світла, від теплого до холодного, а також різні варіанти кольорового світла. Кожна лампа використовує на 80% менше енергії, чим традиційна лампа". Світло можна плавно зменшувати й збільшувати, перебудовувати його колір і все це можливо з будь-якого місця, де ви можете ввійти в Інтернет. Крім того, додаток містить ряд "рецептів" від експертів по світлу, які допоможуть вам краще концентруватися або розслаблятися. Система Hue сумісна з рядом інших пристроїв для "інтелектуального будинку", зокрема з вимикачами Belkin MeMo.

Мінімальний набір (Hue Starter Pack) містить бездротовий хаб і три лампи, додаткові лампи й світильники можуть здобуватися окремо.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Iris

Розроблена англійською компанією AlertMe система тепер продається в США як коробкове рішення. Є кілька варіантів, що розрізняються составом і призначенням. Це набір системи безпеки для початківців, що включає датчики для дверей, вікон і руху, або набір для енергетики, що містить інтелектуальний термостат, комплект для включення й хаб. Хаб можна купити й окремо, а набори можна сполучати. Із цими системами можна докупити додаткові пристрої – внутрішню або зовнішню камеру, дверний замок, інтелектуальні вимикачі й детектор диму. Всі пристрої працюють через хаб, що підтримує стандарти Z-Wave і ZigBee, і який, у свою чергу, може через Wi-Fi виходити в Інтернет і, виходить, контролюватися зі смартфона або комп'ютера.

Ubi

Усе, що описано вище, уже можна придбати. Ubi – це те, що можна буде купити в найближчому майбутньому. Цей пристрій одержав фінансування через краудфандинговий проект (проект народного фінансування) Kikstarter і тепер доступно для передзаказу. Ubi – невелика коробочка, мініатюрний комп'ютер, що містить кілька датчиків, що підключається до Wi-Fi-мережі й управляється голосовими командами. У цей час (проект поки не завершений) Ubi може здійснювати пошук в Інтернет, відсилати SMS і email, а також управляти поруч «інтелектуальних пристроїв». Розроблювачі мають намір дуже сильно розширити функціонал Ubi і, оскільки цей комп'ютер працює на базі Android і має відкритий API (інтерфейс для розроблювачів), він може легко обновлятися.

TELETASK

Компанія TELETASK (Бельгія), виробник устаткування для систем Інтелектуальний будинок, реалізує це встаткування використовуючи технологію MICROS+ (Мікрос плюс).

MICROS+

MICROS+ – це друге покоління модулів MICROS. Перше покоління цих пристроїв випускалося з 2001 по 2007 роки. MICROS+ почали випускати з 2008

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

року. При його розробці були враховані всі нюанси й досвід експлуатації попереднього модуля MICROS. Із застосуванням пристроїв MICROS і MICROS+ у світі виконано вже більше 12000 проектів різного профілю й розміру.

Для реалізації системи Інтелектуальний будинок до MICROS+ підключаються різні навантаження, датчики й інженерні системи. Також до неї підключаються пристрої керування: від звичайних вимикачів до сенсорних екранів і комп'ютера. Ви зможете управляти системою Інтелектуальний будинок на базі MICROS+ за допомогою мобільних пристроїв.

Система Інтелектуальний будинок на базі модуля MICROS+ дозволить:

– управляти світлом, опаленням і теплими підлогами, вентиляцією й кондиціонуванням, шторами й жалюзі, воротами й хвірткою, поливом, системою антиоледеніння, запуском генератора й інших інженерних систем і електричних навантажень;

– контролювати й запобігати аварійні ситуації, пов'язані із протіканнями, відключенням електропостачання, аварією казана, стежити за рівнем палива в баку резервного генератора й т.д.;

– здійснювати функцію охоронної сигналізації;

– управляти аудіо/відео технікою й системою Мультирум;

– програмувати систему Інтелектуальний будинок під свої бажання;

– заощаджувати кошти при експлуатації замиського будинку й квартири;

– створювати комфортні, зручні для життя сценарії для себе й своєї родини.

Модуль MICROS+ може встановлюватися в будинок як індивідуально, так і як центральний модуль системи TELETASK. Він є модулем, до якого можливо підключати інші пристрої системи Інтелектуальний будинок, вироблені компанією TELETASK. Для цього необхідно організувати мережу AUTOBUS.

MICROS+ – це добре продуманий модуль. TELETASK створив його спеціально для застосування у квартирах і замських будинках. У єдиному корпусі MICROS+ поєднує чудовий функціонал, що дозволяє створити комфорт у

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

будинку. MICROS+ зручний у монтажі, налаштуванні й програмуванні. Установивши його у свій будинок, Ви дійсно одержите комфорт для себе й своєї родини. А функції енергозбереження, закладені в MICROS+, дозволять насолоджуватися комфортом і зручністю, і при цьому заощаджувати гроші. Функцію енергозбереження високо оцінили власники замських будинків.

TDS10012 – центральний елемент MICROS+

Центральний елемент оснащений 2-мя шинами AUTOBUS. Підключення до центрального елемента виробляється за допомогою USB або Ethernet. Ethernet-з'єднання може бути підключене до локальної мережі. 8 виходів 0-10 В базово встановлено в модулі. На кожен шину AUTOBUS може бути підключене до 31 пристроя TELETASK. Кількість шин AUTOBUS можна розширити до 4-х за допомогою модуля для розширення мережі. Максимальне число модулів, що підключаються, зростає до $4 \times 31 = 124$ шт.

Застосування: проекти домашньої автоматизації з максимальним навантаженням в 1500 датчиків і пристроїв (вхідних і вихідних).

Характеристики

Виходи

Обладнаний 24-мя замінними реле:

- $I_N = 10 \text{ A} / 250 \text{ VAC} \cos \varphi 0,7$.
- $I_{NI} = 80 \text{ A mp}$ (20 мс) (примітка: пускові струми).

Обладнаний 8-ю виходами 0-10 В для керування диммером. Максимальне навантаження: 500 виходів (загальна кількість реле, диммерів і моторів – 500)

Входи

Обладнаний:

- 32 цифрових входи (для «сухих контактів»);
- 2 аналогових входи (для датчиків температури, освітленості й вологості від TELETASK);
- 2 шини AUTOBUS (з можливістю розширення до 4-х за допомогою TDS10202 – модуль для розширення мережі);

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- USB-роз'єм;
- Ethernet-роз'єм.

На кожен шини AUTOBUS може бути підключене до 31 модуля (один модуль може використовувати більше 1 технічної адреси).

Максимальна довжина шини AUTOBUS 1000 м (близько 150 м без додаткового блока живлення, залежно від підключених пристроїв).

Системні обмеження

Системні обмеження залежать від поточної версії ПЗ:

- 500 місцевих сценаріїв;
- 50 загальних сценаріїв;
- 50 кімнат;
- 50 синхронізованих місцевих сценаріїв;
- 50 сенсорів;
- 50 аудіо зон;
- 250 очевидних функцій;
- 250 функцій таймера;
- 250 функцій вентилятора;
- 250 функцій процесу;
- 500 розкладів;
- 500 прапорів;
- 500 функцій « Якщо-потім-після»;
- 500 повідомлень і сигналів тривоги;
- 500 функцій стану;
- 500 чип-карт.

Обмеження таймера:

- Функція вентилятора: макс. 7200 сек.
- Функції часу: макс. 7200 сек.
- Моторні функції: макс. 7200 сек.
- Синхр. місцеві сценарії: макс 7200 сек. після кожного кроку.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Живлення:

- Напруга: 100-250 VAC.
- Частота: 50/60 ГЦ.
- 45 Вт повне навантаження (макс. споживання 65 Вт).
- Власне споживання: 0,2 А – 1,35 А.

Налаштування: AUTOBUS резистор Убудований у плату MICROS+.

Перезавантаження: Кнопка «Reset»: перезавантаження центрального елемента.

IP-адреса «SW2»: відправлення IP-адреси на комп'ютер (тільки у випадку Ethernet-підключення).

Монтаж

Рекомендується встановлювати на плоску поверхню на рівні очей.

Розділяйте слабкострумові (вхідні) і силові (вихідні) кабельні хвости:

- Слабкострумові: через прямокутний отвір у нижньому лівому куті.
- Силові: через круглі отвори на задній і нижній стінках.

Підключення:

– Максимальний перетин кабелю 1,5 мм² (рекомендується сечення, що, 0,5 – 0,8 мм²)

– Аналогові входи гвинтові клеми, Що Від'єднуються.

Модуль для розширення мережі TDS 10202 (опціонально).

– Використовується спеціальний кабель.

– USB-B з'єднання для прямого підключення до ПК.

– Ethernet. Підключається в локальну мережу. Використовується роз'єм RJ-45 і кабель cat 5e.

– Живлення. Використовується шнур живлення (у комплекті з MICROS+).

– Заземлення. Завжди підключається основна клему заземлення до основної шини заземлення вашої електромережі. Використовуйте гвинтовий затискач у правому нижньому куті.

– Споживання: 0.3 А макс.

– Габарити: 450 Ш x 365 У x 80 Г (мм)

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Мережа AUTOBUS

MICROS+ поєднує в єдиному корпусі чудовий функціонал. Але якщо потрібен більший функціонал від TELETASK, те MICROS+ може бути використаний як центральний (шиноутворюючий) елемент системи.

Шина/мережа AUTOBUS формується MICROS+ і має наступні системні межі:

– 4 підмережі (Autobus 1-4), до кожної з яких може бути підключене до 31 модуля TELETASK. Разом до MICROS+ може бути підключено 124 модуля;

– довжина кожної підмережі Autobus 1-4 може становити до 1000м. Загальна довжина мережі/шини AUTOBUS може становити до 4000 метрів;

– підмережі Autobus 1 і 2 формуються MICROS+ самостійно, а підмережі Autobus 3-4 формуються за допомогою розширника мережі TDS10202.

Фізично шина AUTOBUS являє собою екранований мідний кабель, що складається із двох живильних і двох інформаційних провідників: 2x1,0+2x2x0,22. Інформаційна пара є кручений парою – Cat5.

Живлення мережі AUTOBUS – 12V DC (постійної напруги). Блок живлення 12V уже встановлений в MICROS+ і для підмережі Autobus 1-2 окремі блоки живлення не потрібно.

Для програмування мережі TELETASK центральний елемент системи – MICROS+ підключається до комп'ютера й за допомогою програмного забезпечення виконується налаштування системи.

Кабель TELETASK

MICROS+ поєднує в єдиному корпусі чудовий функціонал. Але якщо потрібен більший функціонал від TELETASK, те MICROS+ може бути використаний як центральний (шиноутворюючий) елемент системи.

Шина/мережа AUTOBUS формується модулем MICROS+. Фізично шина являє собою екранований мідний кабель, що випускається TELETASK, що складається із двох живильних і двох пара інформаційних провідників: 2x1,0+2x2x0,22. Інформаційна пара є кручений парою – Cat5.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

TELETASK випускає кабель для прокладки шини AUTOBUS, бухтами по 100м. Також кабель TELETASK для зручності монтажу випускається в гофротрубе, діам. 20мм.

Після завершення монтажу модуля MICROS+ (або системи TELETASK у цілому) необхідно виконати програмування й пусконаладку.

За допомогою ПЗ виробляється налаштування системи. Як правило, початкове програмування MICROS+ виробляється в офісі розробника Інтелектуальних домів на базі технології TELETASK. На об'єкті виконується його доведення, підлаштування під об'єкт.

Для первісного програмування необхідно продумати – якого функціонала Ви хочете одержати від MICROS+. Чим точніше визначитеся з функціями й сценами керування, тим ефективніше буде робота інженерів по програмуванню й налаштуванню. Необхідні лише конкретні побажання.

Для остаточного налаштування будинку, обладнаного MICROS+, під бажання краще підготуватися. Насамперед, зрозуміти для себе (і, звичайно, для членів родини) чим буде управляти кожна клавіша кожного вимикача. Це заощадить час сервісного інженера, що буде виконувати налаштування.

Але навіть якщо щось забули продумати, нічого страшного. Оскільки MICROS+ може бути підключений до комп'ютера, і за допомогою ПЗ виконуються налаштування, то в будь-який момент часу є можливість їх змінити. Таким чином, маєте гнучку систему завжди готову підбудуватися під будь-які бажання.

Однак, досвід експлуатації показав, що виконання налаштування, якщо дійсно їх продумали для своєї родини, міняються не занадто часто. Як правило, за два-три місяці проживання в будинку, обладнаному MICROS+, люди звикають до комфорту й переналаштовують систему рідко й мінімально, лише для того, щоб щось поліпшити, зробити більше зручним.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3.2 Розробка структурної схеми

Планування «інтелектуального будинку»

При реалізації інтелектуального дому необхідно все заздалегідь спланувати й визначитися зі своїми цілями. Для цього треба скласти короткий список своїх дій:

– Визначите свої мети. Що ви хочете, контролювати свій будинок по телефоні й перевіряти температуру ванної кімнати з будь-якого куточка планети, або ви трохи більше скромні у своїх бажаннях? Напишіть докладно, що в ідеалі ви хочете одержати. Правильний підхід – починати з малого, але знати, що ви хочете одержати наприкінці, це ключовий фактор успішного завершення проекту. Спробуйте зрозуміти, що вам буде найбільше корисно. Почніть, наприклад, із системи безпеки. Або от кілька інших прикладів.

– Система "Відпустка". Натискання однієї кнопки на клавіатурі й буде відключене опалення, система безпеки встане в режим "Включене", а освітлення буде включатися ввечері в певному порядку для імітації наявності будинку людей. Або режим, при якому, у випадку дзвінка у вхідні двері, на ваш телефон приходять зображення що дзвонить, а ви одержує можливість двосторонньої розмови з ним. І ви можете вирішити, чи відкрити йому дистанційно двері (а потім замкнути після його відходу).

– Виберіть стандарт для автоматизації будинку.

– Визначити фази інсталяції систем. Пристрої для "інтелектуального будинку" роблять життя краще, але якщо ви зважитесь відразу на масштабну інсталяцію пристроїв, то через якийсь час можете вирішити, що чи зря ви за це узялися. Розбийте процес на стадії, переконаєтеся, що всі пристрої, які ви встановили на даному етапі, працюють, і тільки потім приступайте до наступної стадії. Ніж докладніше ви намалюєте схему установки, тим простіше ваше життя буде при проведенні робіт.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– Виберіть систему керування. Окремі рішення дозволяють одержати дистанційний доступ до керування.

– Установіть комп'ютер, керуючий системою. Цей комп'ютер повинен працювати в режимі 24/7, тому виберіть для нього місце, де він не буде нікому заважати.

– Установіть пристрої й модулі домашньої автоматизації. Кількість і список пристроїв і керуючих модулів залежить від вашого плану. Їх багато, ми тут не будемо зупинятися на конкретних пристроях, для керування, скажемо, гаражними воротами можна використовувати універсальний модуль. Є модулі для ламп, або стінні вимикачі, більшість із них підключається дуже легко. Про деякі пристрої ми розповімо нижче.

– Для економії електрики використовуйте додаткові датчики, які дозволять, наприклад, системі автоматично виключати світло, коли ви виходите з кімнати.

– Там, де можливо, використовуйте дротові модулі типу вимикачів, а не модулі, що вбудовуються. Це дозволить вам управляти світлом і пристроями без використання системи керування інтелектуальним будинком, якщо щось у ній зламається.

Майте на увазі, що автоматизація вимагає розуміння того, що ви робите, особливо коли ви маєте справу з електрикою. Неправильні дії можуть бути небезпечними для вас.

Основні складові інтелектуального будинку

Опалення, вентиляція, кондиціонування:

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

Зв'язок:

- телефонний зв'язок внутрішня (інтерком) і локальні АТМ;

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- 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;
- 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 витoku водопроводу й витoku газу;
- управління ландшафтними водяними системами (фонтани, водоспади);
- управління температурою води у ваннах і басейнах (термостатування);

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- 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.

Структурна схема зображена на рисунку 3.1. 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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– 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 кондиціонування.

– Управління системою опалення.

– Управління сауною, басейном.

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 за обраним алгоритмом.

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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– "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.

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 тепlopостачання. Досвід показує, що економія експлуатаційних витрат у цьому випадку може досягати 15-20%.

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 воєдино. 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.



Рисунок 3.1 – Структурна схема системи

Важливе питання: яка надійність таких систем, що поєднують безліч «інтелектуальних» пристроїв, наскільки можна бути впевненим, що після того, як ви скомандуєте йому через свій смартфон закрити входні двері, замок дійсно закриється? Є чи в такій системі «захист від дурня»?

Всі системи в будинку, такі як світло, опалення й безпека, є підсистемами єдиної системи автоматизації. Достоїнство цієї системи полягає в тому, що вона забезпечує дуже надійний контроль і інтеграцію всіх підсистем. При цьому майже завжди, коли система автоматизації виходить із ладу, підсистеми продовжують працювати. Це створить для вас тільки деяка незручність. Наприклад, ви не зможете одночасно виключити 20 ламп освітлення натисканням однієї кнопки, але ви однаково зможете це зробити поодиноці й звичайним способом.

Відзначимо, що система "інтелектуального будинку" використовує зворотний зв'язок зі своїми підсистемами й одержує сигнал, якщо щось пішло не так. При цьому на телефон користувача приходять повідомлення й він може прийняти рішення, що йому робити в цьому випадку.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Загальна схема роботи системи така: центральний мікроконтролер (або комп'ютер) приймає сигнали від командних пристроїв, потім передає ці сигнали виконавчим модулям і системам у будинку. Виконавчі модулі й системи одержують команди по електромережі, по інфрачервоному або радіоканалу й включають або вимикають відповідні пристрої: освітлення, систему охорони, кондиціонування повітря, опалення, подачу води й т.д.

Введемо ряд понять, які використовуються при побудові інтелектуальних будинків:

1. Пульти управління. Служить для виконання основних системних функцій: постановки на охорону, зняття з охорони, управління пристроями домашньої автоматики, подачі тривоги.

2. Бездротова клавіатура. Пристрій для управління 16 кінцевими модулями домашньої автоматики, постановка на охорону однією кнопкою, зняття з охорони + функція подачі тривоги.

3. 4-канальний брелок. Прилад, що виконує 5 функцій (3 режими узяття на охорону, управління кінцевими модулями домашньої автоматики й подача тривоги).

4. Магнітоконтат. Перемикач із пружинними контактами з феромагнітного матеріалу, призначений для контролю стану вікон і дверей. Спрацьовує при їхньому розмиканні.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

5. Контрольна панель – «мозок» розумного будинку. Центральний пристрій, скануючий й обробляючий радіосигнали із всіх пристроїв.

6. Датчик витоку газу. Служить для визначення витоку газів. Звичайно має сирену й можливість автоматичного припинення тривоги при падінні концентрації газу нижче встановленого значення.

7. Датчик затоплення. Служить для визначення протікання води в приміщенні. Найчастіше має затримку для попередження помилкових спрацьовувань, викликаних підвищеною вологістю.

8. Детектор диму. Служить для виявлення диму й має внутрішню сирену.

9. Акустичний датчик розбиття скла. Пристрій, що визначає розбиття скла у великому діапазоні частот. Розпізнає розбиття навіть через перешкоди.

10. Бездротова сирена. Служить для оповіщення про спрацьовування того або іншого датчика й має пам'ять тривоги.

11. Пасивний інфрачервоний оповіщувач. Датчик руху з високим ступенем захисту від перешкод і контролем нижньої зони.

12. ПЧ – оповіщувач «Штора» з аналізом вектора руху. Пристрій, призначений для контролю руху з вузькою зоною виявлення. Наприклад, направивши датчик на вікно, користувач спокійно може переміщатися по кімнаті.

13. Інтерфон. Пристрій, призначений для реалізації голосного зв'язку між кімнатами.

14. Бездротовий репітер. Пристрій для посилення площі радіопокриття системи «інтелектуального будинку».

15. Одноканальний брелок. Брелок, що виконує одну з функцій. Як правило, використовується як тривожна кнопка.

Пульти дистанційного управління

Пульт дистанційного управління можна управляти не тільки телевізором і домашнім кінотеатром. З його допомогою легко активувати будь-яку функцію «Інтелектуального будинку»: задати різні сценарії освітлення, відкрити й закрити ворота, поставити будинок на охорону й т.д.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Стационарні комп'ютери, ноутбук або кишеньковий ПК

Будь-який комп'ютер легко інтегрується з "інтелектуальним" будинком. Він може виконувати роль сервера системи. Це дає можливість візуалізованого управління, а також збору даних для статистики. А якщо об'єднати в мережу стаціонарний комп'ютер з кишеньковим, останній можна використовувати як бездротової, інтерактивний пульт управління.

Програми для управління «інтелектуальним» будинком можна встановити на робочому комп'ютері в офісі або на ноутбук, що буде супутником у будь-яких поїздках, і де б ви не були, у відрядженні або на відпочинку, ви завжди зможете стежити за станом свого будинку. З віддаленого комп'ютера, з можливістю виходу в інтернет, можна передати будь-які вказівки, наприклад, дати будинку команду підготуватися до вашого повернення додому – включити систему "клімат-контролю", наповнити ванну.

Сенсорні панелі

Сенсорні панелі рятують власника "інтелектуального" будинку від необхідності запам'ятовувати численні кнопкові комбінації. Панель дозволяє бачити на екрані всі можливості системи й дає доступ до зміни заданих параметрів. І все це з використанням кольорових зображень різних зон і кімнат вашого будинку.

Сенсорні панелі можуть бути переносними або вмонтованими в стіну. Досить легкого торкання, щоб активувати систему. Звичайно вони мають ємну пам'ять і здатні зберігати до декількох сотень тисяч простих команд.

Технології E-mail і SMS

Інший варіант – використовувати для віддаленого управління електронну пошту. При необхідності система може регулярно посилати хазяїнові будинку листа зі звітом про свій стан.

Перевага "інтелектуального" будинку можна розписувати довго. Складно не погодитися з тим, що вони роблять життя комфортнішим й безпечнішим, заощаджують наш час, рятуючи від непотрібних турбот, і бережуть наші нерви,

						ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			37

захищаючи від усіляких побутових стресів. Однак "інтелектуальний" будинок як і раніше залишається дорогим задоволенням, доступним дуже й далеко не всім.

Але не варто сумувати, якщо ми не може зробити весь наш будинок "інтелектуальним", це не виходить, що треба зовсім відмовитися від елементів домашньої автоматизації. Так, наприклад, небагато "розуму" можливо додати навіть звичайній лампочці. Додати його можна за допомогою лампового модуля. Він включається у звичайну розетку й по електричній мережі може управляти будь-якою лампою в будинку, включеної в патрон через спеціальний перехідник. Ламповий модуль дозволить лампочці міняти ступінь яскравості, запам'ятовувати яскравість при вимиканні для того, щоб відновити її після включення, включатися не різко, а плавно.

Ще один приємний "дріб'язок", здатна полегшити життя, – датчики руху. Вони реагують на переміщення людей і замикають електричне коло. Установивши такі датчики в коридорі, можна позбавити себе від неприємних моментів при вході в будинок, коли, наприклад, руки зайняті сумками, а в темряві треба ще умудритися знайти вимикач. Інший варіант – використовувати датчики в замиському будинку для управління вуличним освітленням.

Такі елементи домашньої автоматизації, звичайно ж, лише малий крок на шляху до "інтелектуального" будинку. Але при цьому вони цілком можуть позбавити нас від ряду проблем і зробити наше життя більше приємним й радісним.

Вимикачі й кнопкові панелі

Найбільш простим засобом управління інтелектуальним будинком є "інтелектуальний" вимикач. Одним натисканням на кнопку можна не тільки включити або виключити, але й змінити яскравість освітлення. Це може бути ціла кнопкова панель, де кожна кнопка відповідає якомусь приладу, або їхньої комбінації. Такі вимикачі дозволяють задавати різні режими, наприклад, "режим сну" у спальні – світло гасне плавно, жалюзі закриваються, злегка знижується температура на кондиціонері, установлюється будильник.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

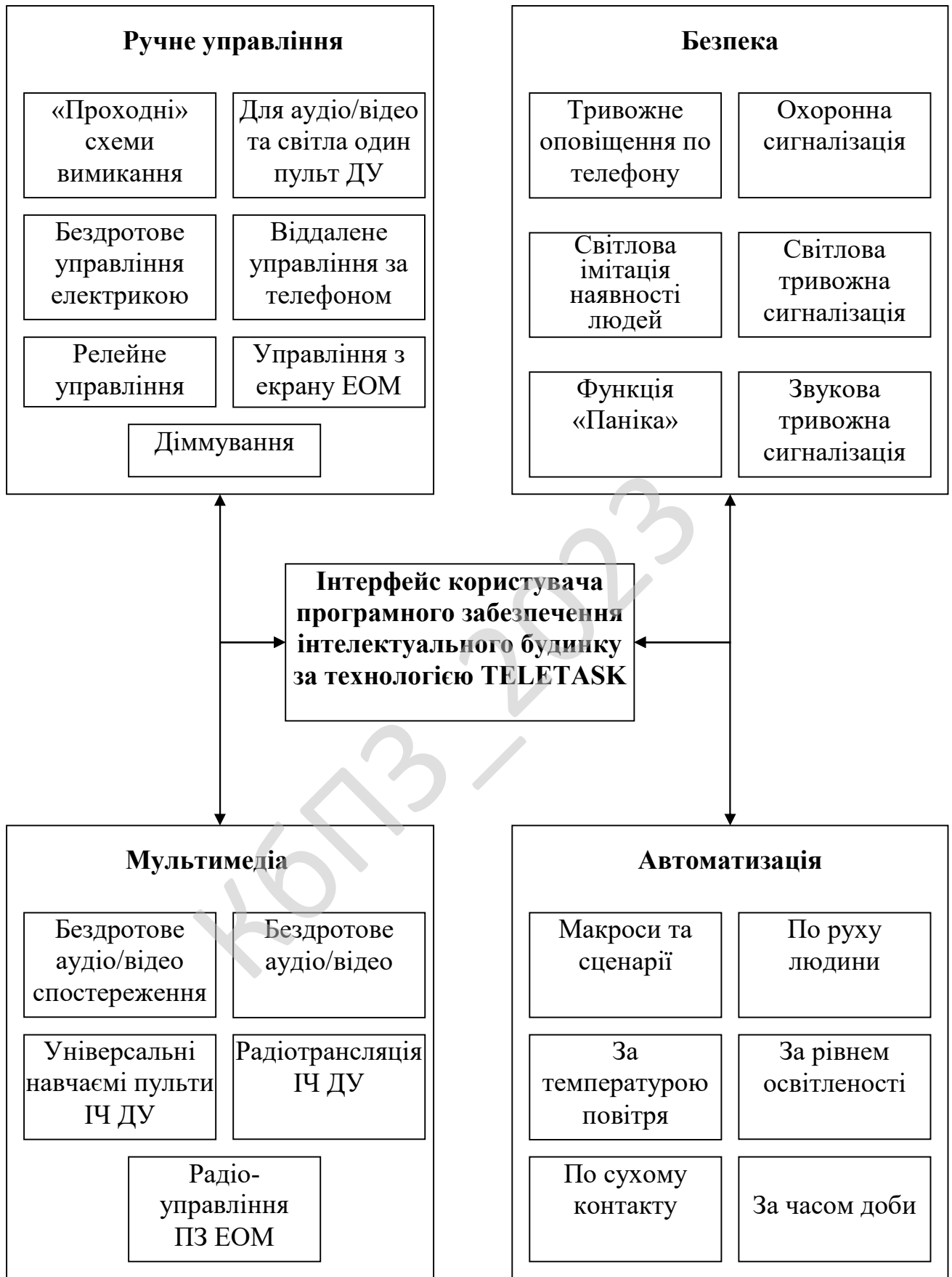


Рисунок 3.2 – Функціональна схема системи

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

– блок функцій, які відповідають за автоматизацію, що дозволяють 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, 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.і.);

– 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 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».

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.

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.

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

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.

Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Виведення головного вікна ПЗ.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- Моніторинг масиву датчиків.
- Датчики детекції руху.
- Датчики відносної вологості повітря.
- Датчики диму.
- Датчики контролю витоку газу.
- Датчики температури.
- Датчики контролю протікання води.
- Моніторинг масиву пристроїв.
- Моніторинг перемикачами.
- Домашній кінотеатр.
- Клімат.
- Газопостачання.
- Водопостачання.
- Освітлення.
- Система безпеки.

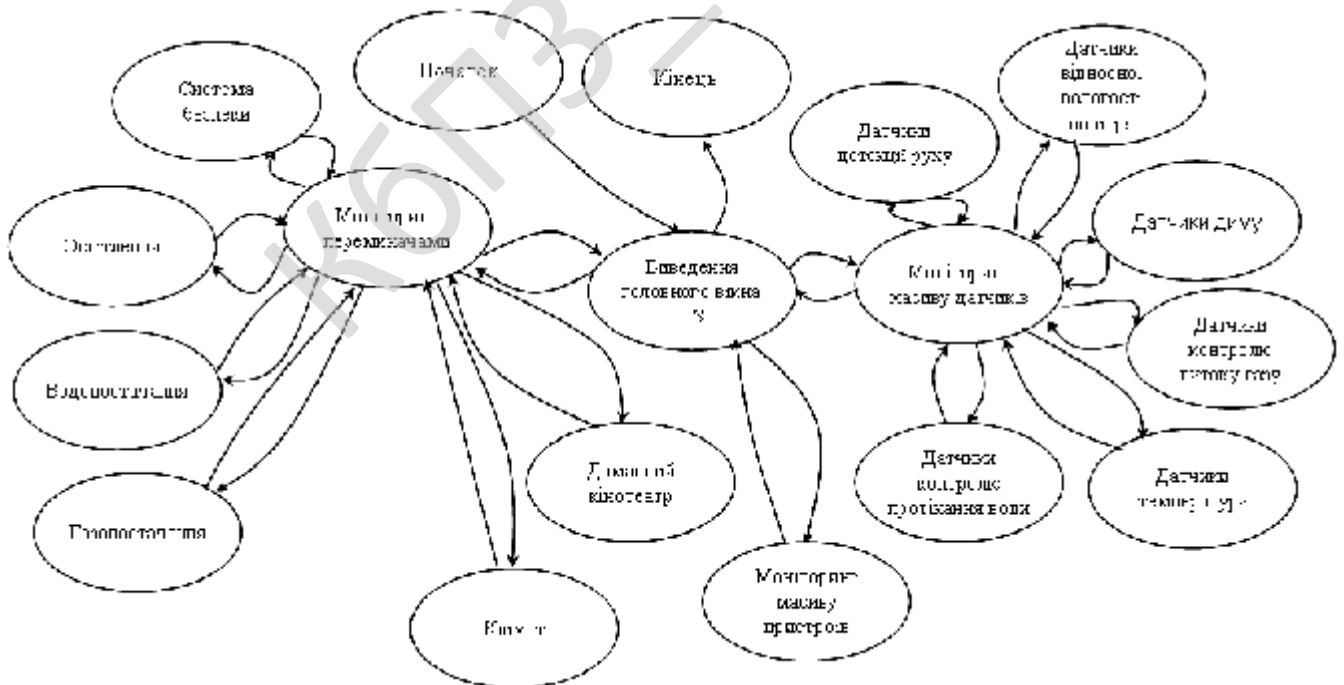


Рисунок 3.3 – Діаграма взаємодії процесів

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

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

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Перед розглядом подробиць схеми роботи програми розглянемо виконані основні напрацювання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування.

Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм.

Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем. Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML.

Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи. Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

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

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

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

– Програмістам які реалізують модулі інформаційної системи.

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

Далі наведемо частини програмного коду програми, які реалізують різні функції програмного забезпечення інтелектуального дому за технологією TELETASK.

```
//вимкнення світла скрізь  
void __fastcall TForm_light_TELETASK::Button17Click(TObject *Sender)
```

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

```

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

```

Керування кліматом:

```
//-----  
void __fastcall TForm_climate_TELETASK::Button3Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,21,18,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button4Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,21,19,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button1Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,22,18,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button2Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,22,19,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button5Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,23,18,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button6Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,23,19,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button7Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,24,18,1);  
}  
//-----  
void __fastcall TForm_climate_TELETASK::Button8Click(TObject *Sender)  
{  
ClientSocket->Send_TELETASK_Command(0,24,19,1);  
}
```

					БКРМ-122.23.0090.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		47

```

//-----
void __fastcall TForm_climate_TELETASK::Button9Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,26,18,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button10Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,26,19,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button11Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,27,18,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button12Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,27,19,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button13Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,28,18,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button14Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,28,19,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button15Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,29,18,1);
}
//-----
void __fastcall TForm_climate_TELETASK::Button16Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,29,19,1);
}
//-----
void __fastcall TForm_climate_TELETASK::TrackBar1Change(TObject *Sender)

```

```

{
ClientSocket->SendLeviton_TELETASK_(0,21,TrackBar1->Position);
}
//-----
void __fastcall TForm_climate_TELETASK::TrackBar2Change(TObject *Sender)
{
ClientSocket->SendLeviton_TELETASK_(0,22,TrackBar2->Position);
}
//-----
void __fastcall TForm_climate_TELETASK::TrackBar3Change(TObject *Sender)
{
ClientSocket->SendLeviton_TELETASK_(0,23,TrackBar3->Position);
}
//-----
void __fastcall TForm_climate_TELETASK::TrackBar4Change(TObject *Sender)
{
ClientSocket->SendLeviton_TELETASK_(0,24,TrackBar4->Position);
}
//-----
void __fastcall TForm_climate_TELETASK::TrackBar5Change(TObject *Sender)
{
ClientSocket->SendLeviton_TELETASK_(0,24,TrackBar5->Position);
}
//-----

```

Керування системою безпеки:

```

//-----
void __fastcall TForm_security_TELETASK::Button14Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,17,18,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button20Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,18,18,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button16Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,19,18,1);
}
//-----

```

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

void __fastcall TForm_security_TELETASK::Button18Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,19,18,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button13Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,15,19,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button19Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,16,19,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button15Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,17,19,1);
}
//-----
void __fastcall TForm_security_TELETASK::Button17Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,18,19,1);
}
//-----
//визначення та виведення стану системи безпеки
void __fastcall TForm_security_TELETASK::StatusTimer(TObject *Sender)
{
ShortString st;
int n;
ClientSocket->Send_TELETASK_StatusQuery();
st=_TELETASK_State->GetStateOnOff(0, 17);
if(st="On") Label_1->Caption="Ввимкнено"; else Label_1->Caption="Вимкнено";
st=_TELETASK_State->GetStateOnOff(0, 18);
if(st="On") Label_2->Caption="Ввимкнено"; else Label_2->Caption="Вимкнено";
st=_TELETASK_State->GetStateOnOff(0, 19);
if(st="On") Label_3->Caption="Ввимкнено"; else Label_3->Caption="Вимкнено";
st=_TELETASK_State->GetStateOnOff(0, 20);
if(st="On") Label_4->Caption="Заблоковано"; else
Label_4->Caption="Відкрито";
n=Getv_TELETASKvariable(17);
}

```

					БКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```

        if(n="0") Label_11->Caption="В нормі"; else Label_11-
>Caption="Спрацьовування";
        n=Getv_TELETASKvariable(18);
        if(n="0") Label_22->Caption="В нормі"; else
        Label_22->Caption="Спрацьовування";
// ввімкнення ліхтаря біля входу в будинок з вказаною яскравістю
void __fastcall TForm_light_TELETASK::torch_onClick(TObject *Sender)
{
    torch->Caption="Ввімкнено";
    Image_torch_on->Visible=true;
    Image_torch_off->Visible=false;
    TrackBar_torch->Enabled=true;
    ClientSocket->Send_TELETASK_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_TELETASK_(0,1,TrackBar_torch->Position);
//встановлення яскравості, вказаної користувачем за допомогою TrackBar-у
}

//-----
//вимкнення ліхтаря біля входу в будинок
void __fastcall TForm_light_TELETASK::torch_offClick(TObject *Sender)
{
    torch->Caption="Вимкнено";
    Image_torch_off->Visible=true;
    Image_torch_on->Visible=false;
    TrackBar_torch->Enabled=false;
    ClientSocket->Send_TELETASK_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_TELETASK::corridor_onClick(TObject *Sender)
{
    corridor->Caption="Ввімкнено";
    Image_corridor_on->Visible=true;
    Image_corridor_off->Visible=false;
    TrackBar_corridor->Enabled=true;
    ClientSocket->Send_TELETASK_Command(0,2,18,1);
    ClientSocket->SendLeviton_TELETASK_(0,2,TrackBar_corridor->Position);
}

```

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```

//-----
//ввімкнення світла у вітальні з вказаною яскравістю
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,3,18,1);
ClientSocket->SendLeviton_TELETASK_(0,3,TrackBar_drawing_room->Position);
}
//-----

//ввімкнення світла на кухні з вказаною яскравістю
void __fastcall TForm_light_TELETASK::kitchen_onClick(TObject *Sender)
{
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
TrackBar_kitchen->Enabled=true;
ClientSocket->Send_TELETASK_Command(0,4,18,1);
ClientSocket->SendLeviton_TELETASK_(0,4,TrackBar_kitchen->Position);
}
//-----

//ввімкнення світла в кабінеті з вказаною яскравістю
void __fastcall TForm_light_TELETASK::cabinet_onClick(TObject *Sender)
{
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
TrackBar_cabinet->Enabled=true;
ClientSocket->Send_TELETASK_Command(0,5,18,1);
ClientSocket->SendLeviton_TELETASK_(0,5,TrackBar_cabinet->Position);
}
//-----

//ввімкнення світла у спальні з вказаною яскравістю
void __fastcall TForm_light_TELETASK::bedroom_onClick(TObject *Sender)
{
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
TrackBar_bedroom->Enabled=true;
ClientSocket->Send_TELETASK_Command(0,6,18,1);
}

```

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

ClientSocket->SendLeviton_TELETASK_(0,6,TrackBar_bedroom->Position);
}
//-----
//ввімкнення світла в дитячій кімнаті з вказаною яскравістю
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,7,18,1);
ClientSocket->SendLeviton_TELETASK_(0,7,TrackBar_baby_room->Position);
}
//-----
//ввімкнення світла у ванній кімнаті з вказаною яскравістю
void __fastcall TForm_light_TELETASK::bathroom_onClick(TObject *Sender)
{
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;
ClientSocket->Send_TELETASK_Command(0,8,18,1);
ClientSocket->SendLeviton_TELETASK_(0,8,TrackBar_bathroom->Position);
}
//-----
//вимкнення світла в коридорі
void __fastcall TForm_light_TELETASK::corridor_offClick(TObject *Sender)
{
corridor->Caption="Вимкнено";
Image_corridor_off->Visible=true;
Image_corridor_on->Visible=false;
TrackBar_corridor->Enabled=false;
ClientSocket->Send_TELETASK_Command(0,2,19,1);
}
//-----
//вимкнення світла у вітальні
void __fastcall TForm_light_TELETASK::drawing_room_offClick(TObject
*Sender)
{
drawing_room->Caption="Вимкнено";
Image_drawing_room_off->Visible=true;
Image_drawing_room_on->Visible=false;
}

```

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53


```

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

//Ввімкнення світла скрізь
void __fastcall TForm_light_TELETASK::Button18Click(TObject *Sender)
{
torch->Caption="Ввімкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,1,18,1);
ClientSocket->SendLeviton_TELETASK_(0,1,TrackBar_torch->Position);
corridor->Caption="Ввімкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,2,18,1);
ClientSocket->SendLeviton_TELETASK_(0,2,TrackBar_corridor->Position);
drawing_room->Caption="Ввімкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,3,18,1);
ClientSocket->SendLeviton_TELETASK_(0,3,TrackBar_drawing_room->Position);
kitchen->Caption="Ввімкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,4,18,1);
ClientSocket->SendLeviton_TELETASK_(0,4,TrackBar_kitchen->Position);
cabinet->Caption="Ввімкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,5,18,1);
ClientSocket->SendLeviton_TELETASK_(0,5,TrackBar_cabinet->Position);
bedroom->Caption="Ввімкнено";
Image_bedroom_on->Visible=true;

```

					БКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

Image_bedroom_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,6,18,1);
ClientSocket->SendLeviton_TELETASK_(0,6,TrackBar_bedroom->Position);
baby_room->Caption="Ввімкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,7,18,1);
ClientSocket->SendLeviton_TELETASK_(0,7,TrackBar_baby_room->Position);
ClientSocket->Send_TELETASK_Command(0,8,18,1);
ClientSocket->SendLeviton_TELETASK_(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;
}

```

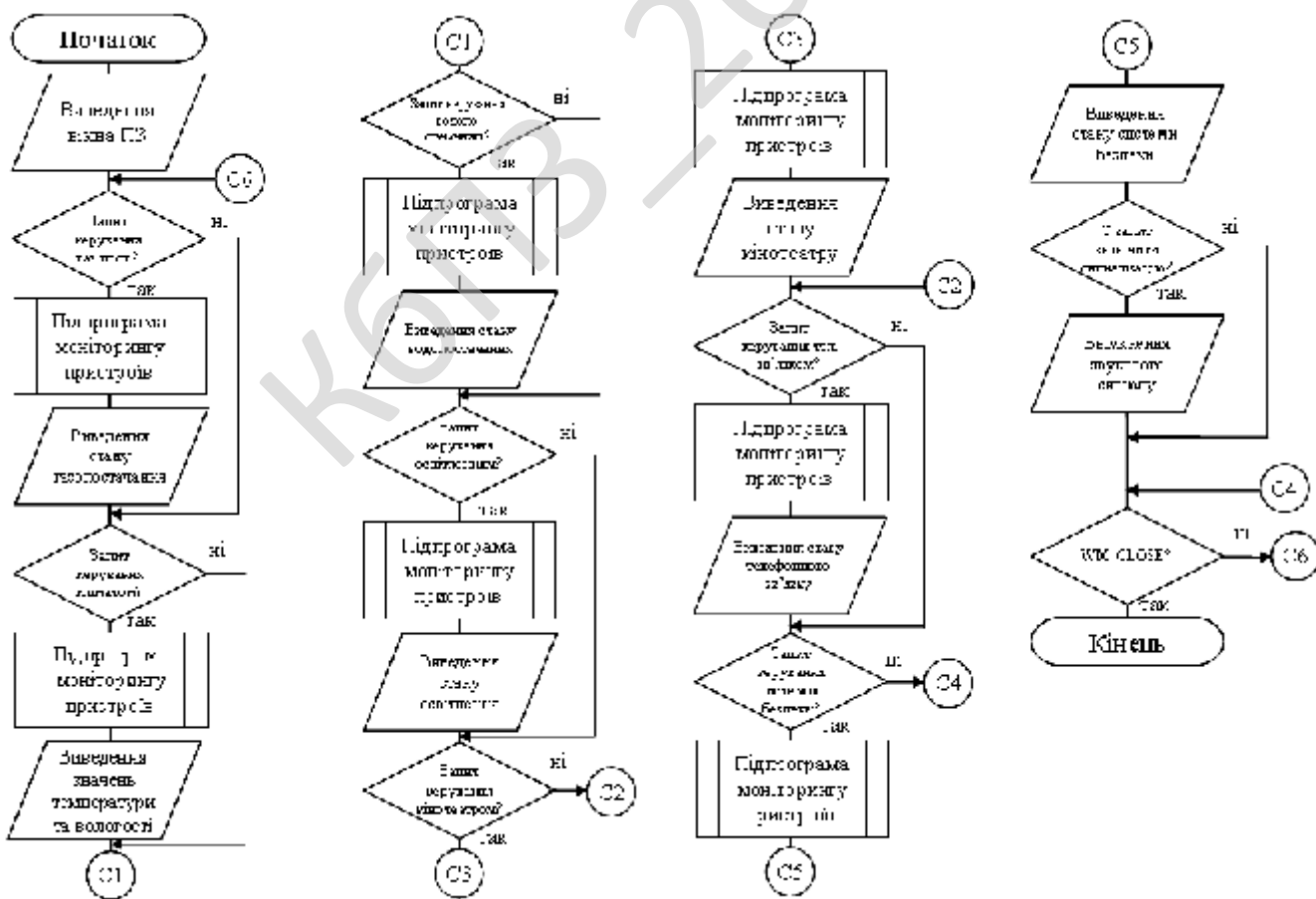


Рисунок 4.1 – Блок схема основної програми

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

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

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

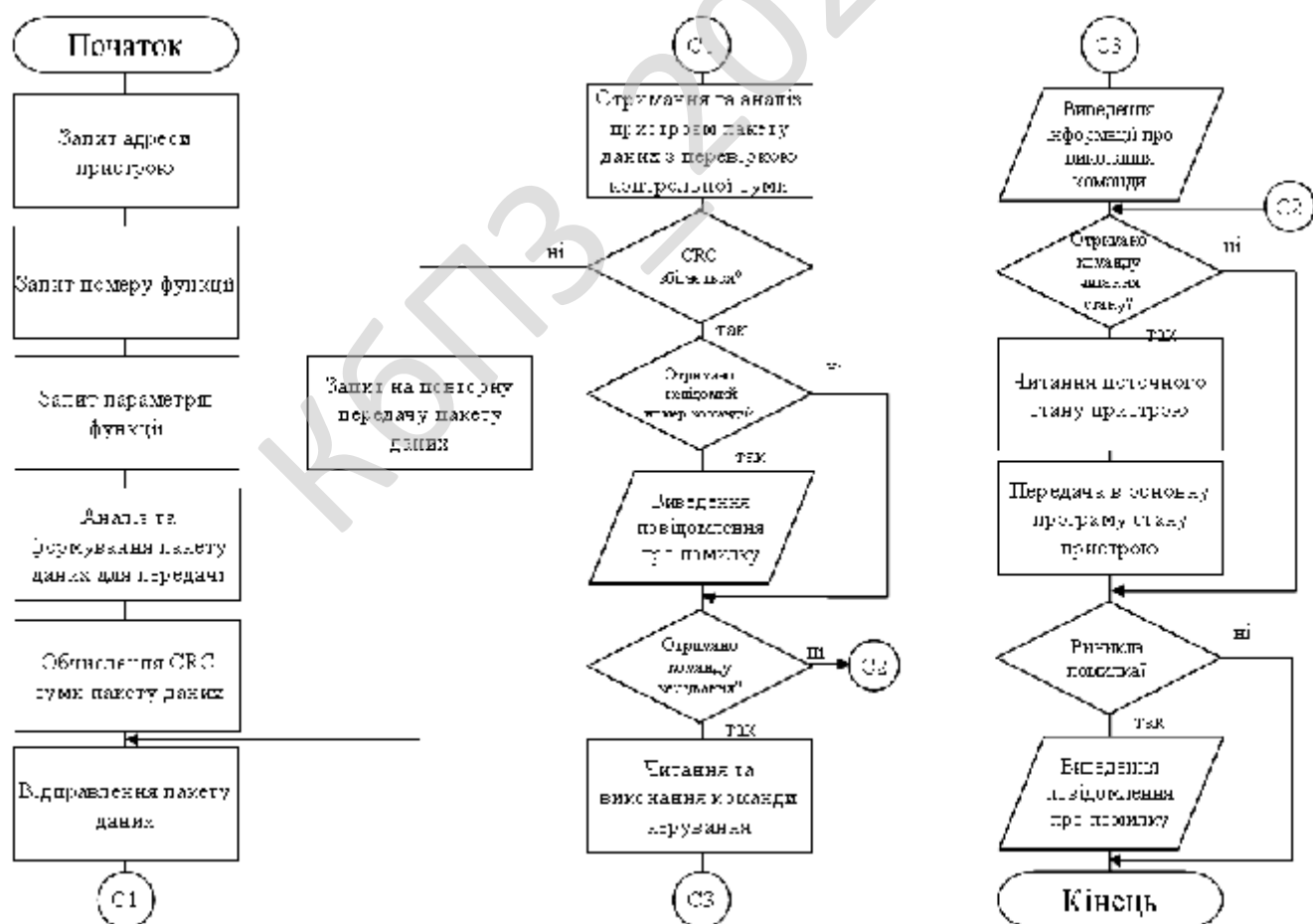


Рисунок 4.2 – Блок схема підпрограм

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Khufu. Khufu – це 64-бітовий блоковий шифр. 64-бітовий відкритий текст спочатку розщеплюється на дві 32-бітові половини, L і R . Над обома половинами й певними частинами ключа виконується операція XOR. Потім, аналогічно DES, результати проходять деяку послідовність раундів. У кожному раунді молодший значущий байт L використовується як вхід S-блоку. У кожного S-блоку 8 вхідних біт і 32 вихідних біта. Далі обраний в S-блоці 32-бітовий елемент піддається операції XOR з R . Потім L циклічно зрушується на число, кратним восьми бітам, L і R міняються місцями, і раунд завершується. Сам S-блок не статичний, він міняється кожні вісім раундів. Нарешті, по закінченні останнього раунду, над L і R виконується операція XOR з іншими частинами ключа, і половини поєднуються, утворюючи блок шифртексту.

Хоча частини ключа використовуються для операції XOR із блоком шифрування на початку й кінці виконання алгоритму, головне призначення ключа – генерація S-блоків. Ці S-блоки секретні, по суті, це частина ключа. Повний розмір ключа алгоритму Khufu дорівнює 512 біт (64 байт), алгоритм надає спосіб генерації S-блоків по ключу.

					VKPM-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми, на якому зображено ті системи інтелектуального дому, якими управляють за технологією TELETASK.

Системи "інтелектуального будинку" або, інакше кажучи, домашньої автоматизації, сьогодні дозволяють робити дуже багато чого – від контролю світла й температури до керування звуком аудіосистем і безпекою будинку й родини, використовуючи для цього звичайні мобільні пристрої.

Система призначена для керування компонентами інтелектуального будинку за технологією TELETASK. Компанія TELETASK (Бельгія) – виробник устаткування для Систем Домашньої Автоматизації (СДА). Областю застосування системи є інтелектуальні будинки. Автоматизація полегшує й поліпшує ваше життя в будинку або квартирі, більше того, вона дозволяє заощаджувати час і гроші. От кілька доводів на користь початку створення свого інтелектуального будинку.

Автоматизація підвищує ефективність. Це відноситься як до ефективності окремих систем у вашому будинку, так і до ефективності всього господарства. Наприклад, оскільки система автоматизації здатна управляти декількома пристроями, ви можете одним натисканням кнопки відключити термостати й виключити світло, коли вирішите поїхати у відпустку. Не треба буде хвилюватися, що ви щось забули зробити.

Перерахуємо компоненти системи інтелектуального дому, якими управляють за технологією TELETASK:

- Клімат-контроль.
- Система водопостачання.
- Домашній кінотеатр.
- Система освітлення.
- Телефонний зв'язок.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

- Система безпеки.
- Система газопостачання.

Саме таким чином організоване основне вікно програми (рис. 5.1).

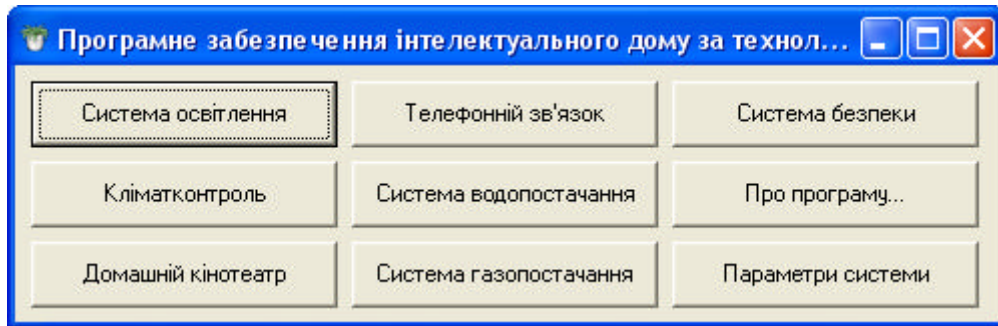


Рисунок 5.1 – Головне вікно програми

На рисунку 5.2 наведено приклад вікна, яке дозволяє керувати системою освітлення у будинку.

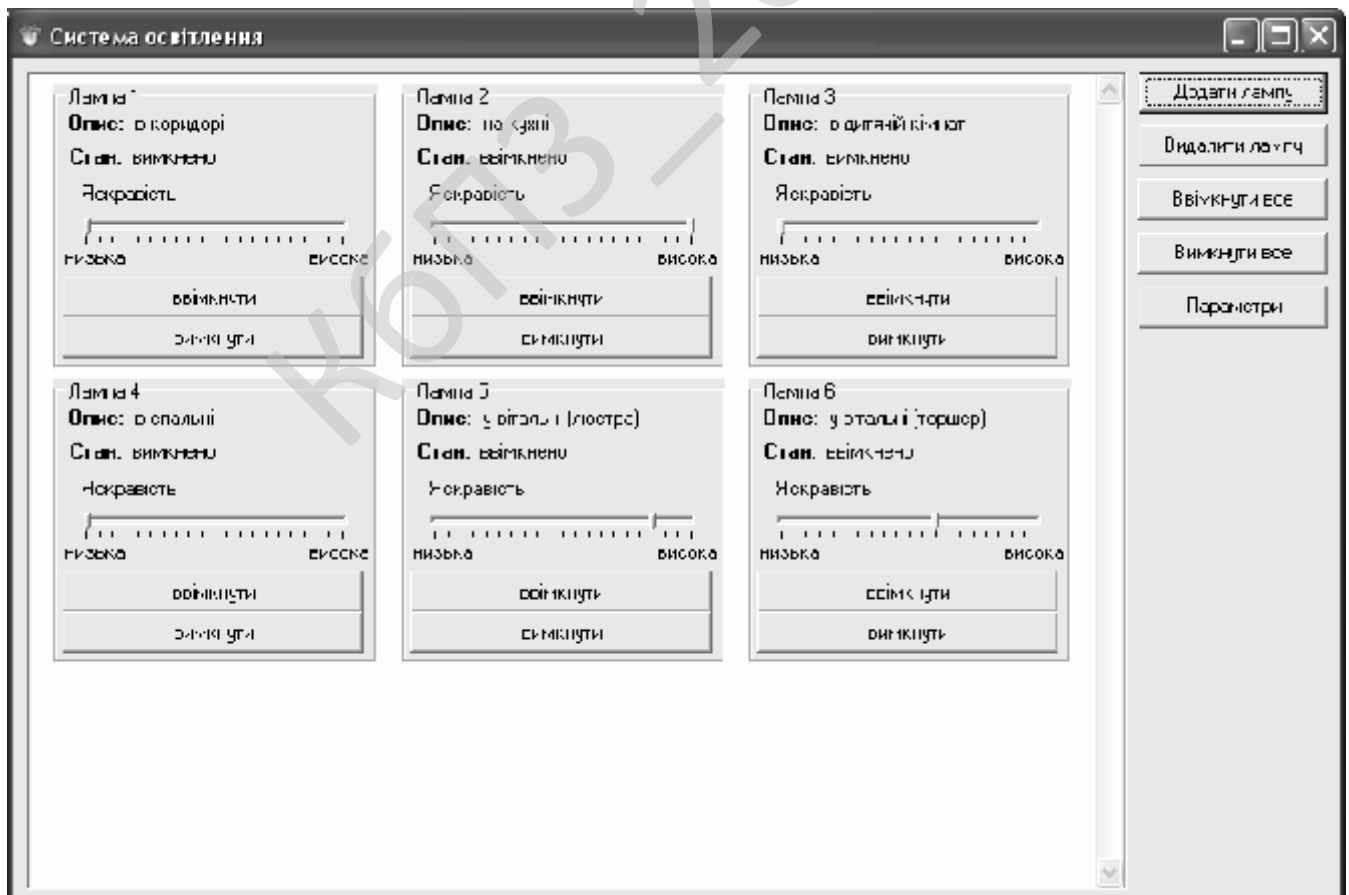


Рисунок 5.2 – Вікно системи освітлення

На рисунку 5.3. наведені дані розробника ПЗ.

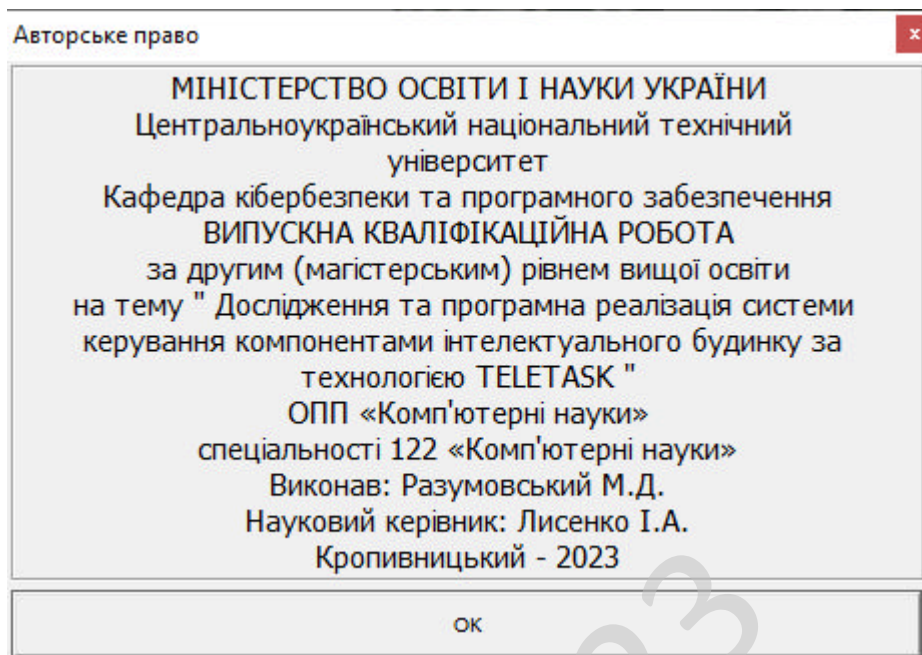


Рисунок 5.3 – Вікно авторського права

Обрано умови розповсюдження – commercial software. Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційних програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації надають відкриті вихідні коди.

Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів.

Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм.

Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

КБПЗ - 2023

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування компонентами інтелектуального будинку за технологією TELETASK.

Метою розробки є дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

Об'єктом дослідження є процес керування компонентами інтелектуального будинку за технологією TELETASK.

Предметом дослідження є методи керування компонентами інтелектуального будинку за технологією TELETASK.

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 керування компонентами інтелектуального будинку за технологією TELETASK.

– Розроблено вітчизняний продукт керування компонентами інтелектуального будинку за технологією TELETASK, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

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 проведене дослідження та виконана програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

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	100
3. Запланований термін розробки, днів	Fpq	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)	–	2
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.	–	100000
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-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 49 = 83 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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	83	Ф 7.1-7.4
Впровадження	13	Д13
Всього	124	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{124 \cdot 1}{60 - 5} = 2,25 \text{ ставки.}$$

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	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	39,49

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:

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

$$\Phi_{\text{др}}^c = \frac{39,49 \cdot 3}{1,2} = 99 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
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	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,25	11124	8343
Продакт-менеджер	0,25	9052	6789
Інженер-програміст	2,25	12200	82350
Інженер - електронщик	0,2	8000	4800
Інженер-системотехнік	0,25	8000	6000
Адміністратор мережі	0,5	8000	12000
Системний програміст	0,25	8000	6000
Дизайнер WEB	0,25	8000	6000
Інженер-верстальник	0,25	8000	6000
Бухгалтер-економіст	0,5	8000	12000
Всього за період розробки	$R_{cn} = 4,95$	-	$\Phi_{роб} = 150282$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{150282}{4,95 \cdot 60} = 506 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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

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 інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 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:

$$B_{y\delta} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

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

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

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

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

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 Комп'ютерторг за 20.10.23 – джерело <http://computorg.ua/ru/price.html>

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 3.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

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
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.

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	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

Примітка: вартість автомобіля Renault Scenic 2006 взята за даними електронного ресурсу, джерело https://auto.ria.com/uk/auto_renault_scenic_33598032.html, складає 143000 грн.

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.

$$Z_o = 506 \cdot 164 / 100 = 629 \text{ грн.}$$

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, %.

$$Z_d = 629 \cdot 10 \cdot 0,01 = 63 \text{ грн.}$$

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, %.

$$C_{oc} = 0,01 \cdot 22(629+63) = 256 \text{ грн.}$$

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, %.

$$G_{ocn} = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

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-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Згідно прийнятих норм на підприємстві $n_{\text{вум}}$ приймаємо 1,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=206$ грн., визначаємо вартість паперу за період розробки:

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

$$З_{M1} = 206 \cdot 1,5 = 309 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 50):

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

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 24 грн./шт., DVD-R box – 39 грн./шт.

$$З_{M2} = 49 \cdot 24 + 1 \cdot 39 = 1215 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_{\gamma}, \quad (7.18)$$

де: $Ц_{\gamma}$ – 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.

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

$$З_M = (309 + 1215 + 1702) / 100 = 32 \text{ грн.}$$

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, %.

$$O_n = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 100$ прим.):

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

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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

$$A_m = 249636 \cdot 3 / (100 \cdot 12) = 624 \text{ грн.}$$

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	2	3
1. Основна зарплата виконавців	$З_o$	629
2. Додаткова зарплата виконавців	$З_\delta$	63
3. Відрахування на соціальні потреби	C_{oc}	256
4. Загальногосподарські витрати	Γ_{ocn}	94
5. Витрати на матеріали	$З_M$	32
6. Освоєння нових операційних систем, мов програмування	O_n	94
7. Амортизація основних фондів	A_m	624
8. Повна собівартість програмного забезпечення	C_n	1792
9. Плановий прибуток	Π_p	896
10. Ціна підприємства $C_n = C_n + \Pi_p$	C_n	2688
11. Податок на додану вартість $\Pi_{ДВ} = 0.01 \cdot H_{об} \cdot C_n$	$\Pi_{ДВ}$	537,6
12. Відпускна ціна програмної продукції $C = C_n + \Pi_{ДВ}$	C	3225,6

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

$$C_n = З_o + З_\delta + C_{oc} + \Gamma_{ocn} + З_M + O_n + A_m. \quad (7.21)$$

$$C_n = 629 + 63 + 256 + 94 + 32 + 94 + 624 = 1792 \text{ грн.}$$

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

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, %.

$$P_p = 0,01 \cdot 50 \cdot 1792 = 896 \text{ грн.}$$

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 7.10.

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3226
Всього капітальних витрат	–	3226

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 технічне обслуговування)	Z_p	32208	12078
2. Витрати на електроенергію	$Z_{ел}$	276	58
3. Витрати на амортизацію	$Z_{ам}$	0	807
Всього витрат за рік	I	32484	12943

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 кількість профілактичних годин робіт зменшилася з 240 годин на рік до 90 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 100 \cdot 1,1 \cdot 1,22 = 32208 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 90 \cdot 100 \cdot 1,1 \cdot 1,22 = 12078 \text{ грн}.$$

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 ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,5 \cdot 240 \cdot 2,3 = 276 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,5 \cdot 50 \cdot 2,3 = 58 \text{ грн}.$$

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	–	3226	–	806,5
Всього відрахувань	-	–	3226	–	806,5

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:

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

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

$$E_e = (2688-1792) \cdot 100 - (0,05 \cdot 2043800 + 0,4 \cdot 199177 + 0,25 \cdot 37031 + 0,1 \cdot 100000 + 0,2 \cdot 143000) \cdot 3/12 = 32170 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{479208}{(2688-1792) \cdot 100 \cdot 12 / 3} = 1,3 \text{ років.}$$

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.	100
2. Повна собівартість розробленої програми	Грн.	1792
3. Ціна розробленої програми	Грн.	2688
4. Плановий прибуток від реалізації розробленої програми	Грн.	896
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	89600
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	32170
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	18735
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,16

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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.

$$E_{cn} = (32484 - 12943) - 0,25 \cdot 3226 = 18735 \text{ грн.}$$

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

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{3226}{32484 - 12943} = 0,16 \text{ року.}$$

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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

– 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, хобі 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.

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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

– Забезпечувати безпечні і нешкідливі умови праці для працівників, використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби клектинго 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 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами. Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам. Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань. Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

8.3 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За пожежо-вибухонебезпеку приміщення відноситься до класу В. В таблиці 8.2 наведена загальна характеристика приміщення щодо вибухопожеженобезпеки та важкістю робіт.

Таблиця 8.1 – Загальна характеристика приміщення щодо вибухопожеженобезпеки та важкістю робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – П	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а.....до 139 Вт/м ² 1б.....до 140-174 Вт/м ² Клас умов праці – оптимальний

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному

приміщенні, відноситься до категорії I-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [1].

За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [2] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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;

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [4, 10, 11].

8.5 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток $63 \cdot 63 \cdot 6$ мм., (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною $L=2$ м., та горизонтальний електрод – металева полоса з перетином $60 \cdot 5$ мм. Напруга – $220/380$ В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

заземлювачами (електродами) $A=2$ м. Глибина закладення горизонтального контуру заземлення $t=0,6$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,6+2/2=1,6 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho_2 = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [12];

$\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [12].

Еквівалентний діаметр вертикального електрода (кутка) [12]:

$$D_e = 0,95 \cdot K = 0,95 \cdot 63 = 59,85 \text{ мм.} = 0,0598 \text{ м.}$$

де $K = 63$ мм. – розмір металевих кутків (задан).

Відношення $A/L = 2/2 = 1$

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [12]:

$$\begin{aligned} R_0 &= 0,366(\rho/L)[\lg(2L/D_e) + (1/2)\lg((4T+L)/(4T-L))] = \\ &= 0,366(54,5/2)[\lg(2 \cdot 2/0,0598) + (1/2)\lg((4 \cdot 1,6+2)/(4 \cdot 1,6-2))] = \\ &= 19,6 \text{ Ом.} \end{aligned}$$

Визначаємо коефіцієнт екранування вертикальних електродів $K_{ев} = 0,62$ при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [12].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{3H} = 4$ Ом:

$$N = R_0 / (K_{ев} R_{3H}) = 19,6 / (0,62 \cdot 4) = 7,8 \approx 8 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

КБПЗ_2023

					VKPM-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування компонентами інтелектуального будинку за технологією TELETASK.

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 керування компонентами інтелектуального будинку за технологією TELETASK.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем керування компонентами інтелектуального будинку за технологією TELETASK.
- Досліджена система керування компонентами інтелектуального будинку за технологією TELETASK.
- На основі отриманих результатів досліджень створена програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування компонентами інтелектуального будинку за технологією TELETASK.

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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

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

При 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 Khufu.

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 18735 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,16 роки.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Разумовський М.Д. Дослідження та програмна реалізація системи керування компонентами інтелектуального будинку за технологією TELETASK // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. 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.
3. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.
4. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.
5. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.
6. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
7. 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.
8. 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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties».

					BKPM-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

17. 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.

18. 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.

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

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.*

25. 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. – 2024. – P. 61-78.

26. 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.

27. 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.

28. 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.

29. 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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

30. 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.

31. 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

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

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

38. 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).

39. 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

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

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

42. 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.

43. 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.

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

45. 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.

46. 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.

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. - 2016. - С. 121-127.

48. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. - Випуск 2 (46) - Х.: ХУПС - 2016. - С. 146-149.

49. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

50. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

51. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

52. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

					ВКРМ-122.23.0090.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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-122.23.0090.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Разумовський М.Д.				Літ.	Аркуш	Аркушів
Перевірів	Лисенко І.А.				М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи керування компонентами інтелектуального будинку за технологією TELETASK.

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. № 37-13 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 керування компонентами інтелектуального будинку за технологією TELETASK.

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-122.23.0090.00.00.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 керування компонентами інтелектуального будинку за технологією TELETASK;
- 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-122.23.0090.00.00.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-122.23.0090.00.00.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 повинна бути розглянута пожежна безпека.

					ВКРМ-122.23.0090.00.00.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	– 103 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 15.12.2023 р.

					ВКРМ-122.23.0090.00.00.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

_____ Лисенко І.А

*Дослідження та програмна реалізація
системи керування компонентами інтелектуального будинку за
технологією TELETASK*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 50

Літера: РП

Кропивницький – 2023 року

Файл CPU_TELETASK_Socket.cpp - протокол _TELETASK_

```

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

#pragma hdrstop

#include "CPU_TELETASK_Socket.h"

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

static inline void ValidCtrCheck(TCPU_TELETASK_Socket *)
{
    new TCPU_TELETASK_Socket(NULL);
}
//#include "OcelotStateUnit.cpp"
//-----
__fastcall TCPU_TELETASK_Socket::TCPU_TELETASK_Socket(TComponent* Owner)
    : TClientSocket(Owner)
{
    _TELETASK_StateChangeNum=new T_TELETASK_StateChangeNum;
    OcelotStateChangeNum=new TOcelotStateChangeNum;
    _TELETASK_State=new T_TELETASK_State;
    OcelotState=new TOcelotState;

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

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

__fastcall TCPU_TELETASK_Socket::~TCPU_TELETASK_Socket(void)
{
    delete _TELETASK_StateChangeNum;
    delete OcelotStateChangeNum;
    delete _TELETASK_State;
    delete OcelotState;
}
//-----
namespace CPU_TELETASK_Socket
{
    void __fastcall PACKAGE Register()
    {
        TComponentClass classes[1] = {__classid(TCPU_TELETASK_Socket)};
        RegisterComponents(" CPU-XA", classes, 0);
    }
}
//-----
void __fastcall TCPU_TELETASK_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_TELETASK_Socket::MyOnDisconnect(System::TObject* Sender,
TCustomWinSocket* Socket)
{
SendBuf("d",1);
FAuth=2;
if (FOnChangeAuthStatus)
    FOnChangeAuthStatus(this,FAuth);
}

//-----
void __fastcall TCPU_TELETASK_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_TELETASK_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(&TELETASK_StateChangeNum->Num,&local_pointer[1],2);
        _TELETASK_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_TELETASK_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_TELETASK_Socket::SendOcelotStatusQuery(void)
{
char buf[3];
buf[0]='q';
memcpy(&buf[1],&OcelotStateChangeNum->Num,2);
SendBuf(buf,3);
}
//-----
void __fastcall TCPU_TELETASK_Socket::SendI0StatusQuery(void)
{
char buf[3];
buf[0]='x';
memcpy(&buf[1],&_TELETASK_StateChangeNum->Num,2);
SendBuf(buf,3);
}
//-----
unsigned short __fastcall TCPU_TELETASK_Socket::GetDataFromUnit(unsigned char
UnitNumber)
{
return OcelotState->GetDataFromUnit(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_TELETASK_Socket::GetIO(unsigned char
UnitNumber,unsigned char Point)
{
return OcelotState->GetIO(UnitNumber,Point);
}
//-----
unsigned char __fastcall TCPU_TELETASK_Socket::GetUnitv_TELETASKer(unsigned char
UnitNumber)
{
return OcelotState->GetUnitv_TELETASKer(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_TELETASK_Socket::GetUnitType(unsigned char
UnitNumber)
{
return OcelotState->GetUnitType(UnitNumber);
}
//-----
unsigned char __fastcall TCPU_TELETASK_Socket::GetUnitsCount(void)
{
return OcelotState->GetUnitsCount();
}
//-----
TDateTime __fastcall TCPU_TELETASK_Socket::GetCPUXADateTime()
{
return OcelotState->GetCPUXADateTime();
}
//-----
unsigned short __fastcall TCPU_TELETASK_Socket::Getv_TELETASKvariable(unsigned
char VarNumber)
{
return OcelotState->Getv_TELETASKvariable(VarNumber);
}

```

```

}
//-----
unsigned short __fastcall TCPU_TELETASK_Socket::GetTimer(unsigned char
TimerNumber)
{
return OcelotState->GetTimer(TimerNumber);
}
//-----
TDateTime TCPU_TELETASK_Socket::GetTimerAsTime(unsigned char VarNumber)
{
return UShortToTime(OcelotState->GetTimer(VarNumber));
}
//-----
TDateTime TCPU_TELETASK_Socket::Getv_TELETASKarAsTime(unsigned char VarNumber)
{
return UShortToTime(OcelotState->Getv_TELETASKariable(VarNumber));
}
//-----
TDateTime TCPU_TELETASK_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_TELETASK_Socket::SetTimeAsVar(unsigned char VarNumber, TDateTime time)
{
Setv_TELETASKariable(VarNumber,TimeToUShort(time));
}
//-----
void TCPU_TELETASK_Socket::SetTimeAsTimer(unsigned char VarNumber, TDateTime
time)
{
SetTimer(VarNumber,TimeToUShort(time));
}
//-----
unsigned short TCPU_TELETASK_Socket::TimeToUShort(TDateTime time)
{
Word Hour, Min, Sec, MSec;
DecodeTime(time, Hour, Min, Sec, MSec);
return Hour*100+Min;
}
//-----
void __fastcall TCPU_TELETASK_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_TELETASK_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_TELETASK_Socket::Setv_TELETASKvariable(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_TELETASK_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_TELETASK_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_TELETASK_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_TELETASK_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_TELETASK_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_TELETASK_Socket::GetInternalProtocolVersion(void)
{
char buf[1];
buf[0]='p';
SendBuf(buf,1);
}
//-----

void __fastcall T_TELETASK_StateChangeNum::Next(void)
{
if (Num==65534)
    Num=1;

else
    Num++;
};
//-----
__fastcall T_TELETASK_StateChangeNum::T_TELETASK_StateChangeNum(void)
{
Num=1;
};
//-----
AnsiString __fastcall T_TELETASK_State::GetStateOnOff(unsigned char house,
unsigned char key)
{
if (map[house][key]==true)
return "On";

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

else
return false;
}
//-----

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

void __fastcall TCPU_TELETASK_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_TELETASK_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;
}
```

К6ПЗ_2023

Файл CPU_TELETASK_Socket.h - бібліотека для файлу CPU_TELETASK_Socket.cpp

```
//-----
#ifndef CPU_TELETASK_Socket
#define CPU_TELETASK_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_TELETASK_StateChangeNum {
public:      // User declarations
unsigned short Num;
void __fastcall Next(void);
__fastcall T_TELETASK_StateChangeNum(void);
};

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

class PACKAGE TCPU_TELETASK_Socket : public TClientSocket
{
private:
    AnsiString FVersion;
    AnsiString FLogin;
    AnsiString FPassword;
    unsigned char FAuth;
    TOcelotStateChangeNum *OcelotStateChangeNum;
    T_TELETASK_StateChangeNum *_TELETASK_StateChangeNum;
    TOcelotState *OcelotState;

    // перевірка правильності введення даних
    void __fastcall SetLogin(AnsiString str);
    void __fastcall SetPassword(AnsiString str);
    void __fastcall Setv_TELETASKersion(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);
// Показчики 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_TELETASK_Socket(TComponent* Owner);
__fastcall ~TCPU_TELETASK_Socket(void);
T_TELETASK_State * _TELETASK_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 GetUnitv_TELETASKer(unsigned char UnitNumber);
unsigned char __fastcall GetUnitType(unsigned char UnitNumber);
unsigned char __fastcall GetUnitsCount(void);
TDateTime __fastcall GetCPUXADateTime();
unsigned short __fastcall Getv_TELETASKvariable(unsigned char VarNumber);
TDateTime Getv_TELETASKarAsTime(unsigned char VarNumber);
TDateTime GetTimerAsTime(unsigned char VarNumber);
void SetTimeAsVar(unsigned char VarNumber, TDateTime time);
void SetTimeAsTimer(unsigned char VarNumber, TDateTime time);
unsigned short __fastcall GetTimer(unsigned char TimerNumber);
// Sending commands
void __fastcall SetIO(unsigned char UnitNumber, unsigned char Point, unsigned
char Stat);
void __fastcall SetDateTime(unsigned char day, unsigned char mon, unsigned
char year, unsigned char hour, unsigned char min);
void __fastcall Setv_TELETASKvariable(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 =
Setv_TELETASKersion};
// 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
```

К6П3_2023

Основна програма

Файл main.cpp основної програми

```

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

//підключення модулів програми
#include "main.h"
#include "light_TELETASK.h"
#include "water_TELETASK.h"
#include "gas_TELETASK.h"
#include "security_TELETASK.h"
#include "climate_TELETASK.h"
#include "phone_TELETASK.h"
#include "tv_TELETASK.h"
#include "about_TELETASK.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_main *Form_main;
//-----
__fastcall TForm_main::TForm_main(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
//відкриття вікна "Система безпеки"
void __fastcall TForm_main::Button3Click(TObject *Sender)
{
Form_security_TELETASK->Show();
}
//-----
//відкриття вікна "Домашній кінотеатр"
void __fastcall TForm_main::Button7Click(TObject *Sender)
{
Form_tv_TELETASK->Show();
}
//-----
//відкриття вікна "Телефонний зв'язок"
void __fastcall TForm_main::Button6Click(TObject *Sender)
{
Form_phone_TELETASK->Show();
}
//-----
//відкриття вікна "Управління освітленням"
void __fastcall TForm_main::Button1Click(TObject *Sender)
{
Form_light_TELETASK->Show();
}
//-----
//відкриття вікна "Про програму..."
void __fastcall TForm_main::Button8Click(TObject *Sender)
{
Form_about_TELETASK->Show();
}
//-----

```

```
//відкриття вікна "Система водопостачання"  
void __fastcall TForm_main::Button4Click(TObject *Sender)  
{  
Form_water_TELETASK->Show();  
}  
//-----  
  
//відкриття вікна "Система клімат-контролю"  
void __fastcall TForm_main::Button2Click(TObject *Sender)  
{  
Form_climate_TELETASK->Show();  
}  
//-----  
  
//відкриття вікна "Система газопостачання"  
void __fastcall TForm_main::Button5Click(TObject *Sender)  
{  
Form_gas_TELETASK->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:      // IDE-managed Components
    TImage *Image1;
    TButton *Button1;
    TButton *Button2;
    TButton *Button3;
    TButton *Button4;
    TButton *Button5;
    TButton *Button6;
    TButton *Button7;
    TImage *Image2;
    TImage *Image3;
    TImage *Image4;
    TImage *Image5;
    TImage *Image6;
    TImage *Image7;
    TImage *Image8;
    TButton *Button8;
    TImage *Image9;
    TLabel *Label1;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button8Click(TObject *Sender);
    void __fastcall Button4Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
    void __fastcall Button5Click(TObject *Sender);
    void __fastcall Button3Click(TObject *Sender);
    void __fastcall Button7Click(TObject *Sender);
    void __fastcall Button6Click(TObject *Sender);
private:      // User declarations
public:      // User declarations
    __fastcall TForm_main(TComponent* Owner);
};
//-----
extern PACKAGE TForm_main *Form_main;
//-----
#endif

```

Файл Project_home_TELETASK.cpp основної програми

```

#include <vcl.h>
#pragma hdrstop
//-----
USEFORM("main.cpp", Form_main);
USEFORM("about_TELETASK.cpp", Form_about_TELETASK);
USEFORM("light_TELETASK.cpp", Form_light_TELETASK);
USEFORM("water_TELETASK.cpp", Form_water_TELETASK);
USEFORM("gas_TELETASK.cpp", Form_gas_TELETASK);
USEFORM("security_TELETASK.cpp", Form_security_TELETASK);
USEFORM("climate_TELETASK.cpp", Form_climate_TELETASK);
USEFORM("phone_TELETASK.cpp", Form_phone_TELETASK);
USEFORM("tv_TELETASK.cpp", Form_tv_TELETASK);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm_main), &Form_main);
        Application->CreateForm(__classid(TForm_about_TELETASK),
&Form_about_TELETASK);
        Application->CreateForm(__classid(TForm_light_TELETASK),
&Form_light_TELETASK);
        Application->CreateForm(__classid(TForm_water_TELETASK),
&Form_water_TELETASK);
        Application->CreateForm(__classid(TForm_gas_TELETASK),
&Form_gas_TELETASK);
        Application->CreateForm(__classid(TForm_security_TELETASK),
&Form_security_TELETASK);
        Application->CreateForm(__classid(TForm_climate_TELETASK),
&Form_climate_TELETASK);
        Application->CreateForm(__classid(TForm_phone_TELETASK),
&Form_phone_TELETASK);
        Application->CreateForm(__classid(TForm_tv_TELETASK),
&Form_tv_TELETASK);

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

```

Файл light_TELETASK.cpp - керування освітленням

```

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

#include "light_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_light_TELETASK *Form_light_TELETASK;
//-----
__fastcall TForm_light_TELETASK::TForm_light_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

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

    ClientSocket->Send_TELETASK_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_TELETASK_(0,1,TrackBar_torch->Position);
    //встановлення яскравості, вказаної користувачем за допомогою TrackBar-у
}
//-----
//вимкнення ліхтаря біля входу в будинок
void __fastcall TForm_light_TELETASK::torch_offClick(TObject *Sender)
{
    torch->Caption="Вимкнено";
    Image_torch_off->Visible=true;
    Image_torch_on->Visible=false;
    TrackBar_torch->Enabled=false;
    ClientSocket->Send_TELETASK_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_TELETASK::corridor_onClick(TObject *Sender)
{
    corridor->Caption="Ввімкнено";
    Image_corridor_on->Visible=true;
    Image_corridor_off->Visible=false;
    TrackBar_corridor->Enabled=true;

    ClientSocket->Send_TELETASK_Command(0,2,18,1);
    ClientSocket->SendLeviton_TELETASK_(0,2,TrackBar_corridor->Position);
}
//-----
//ввімкнення світла у вітальні з вказаною яскравістю
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,3,18,1);
ClientSocket->SendLeviton_TELETASK_(0,3,TrackBar_drawing_room->Position);
}
//-----

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

ClientSocket->Send_TELETASK_Command(0,4,18,1);
ClientSocket->SendLeviton_TELETASK_(0,4,TrackBar_kitchen->Position);
}
//-----

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

ClientSocket->Send_TELETASK_Command(0,5,18,1);
ClientSocket->SendLeviton_TELETASK_(0,5,TrackBar_cabinet->Position);
}
//-----

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

ClientSocket->Send_TELETASK_Command(0,6,18,1);
ClientSocket->SendLeviton_TELETASK_(0,6,TrackBar_bedroom->Position);
}
//-----

//ввімкнення світла в дитячій кімнаті з вказаною яскравістю
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,7,18,1);
ClientSocket->SendLeviton_TELETASK_(0,7,TrackBar_baby_room->Position);
}
//-----

//ввімкнення світла у ванній кімнаті з вказаною яскравістю
void __fastcall TForm_light_TELETASK::bathroom_onClick(TObject *Sender)
{
bathroom->Caption="Ввімкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
TrackBar_bathroom->Enabled=true;

ClientSocket->Send_TELETASK_Command(0,8,18,1);
ClientSocket->SendLeviton_TELETASK_(0,8,TrackBar_bathroom->Position);
}

```

```
}

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

//вимкнення світла у вітальні
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,3,19,1);
}
//-----

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

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

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

//вимкнення світла у дитячій кімнаті
void __fastcall TForm_light_TELETASK::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_TELETASK_Command(0,7,19,1);
}
}
```

```

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

//Ввимкнення світла скрізь
void __fastcall TForm_light_TELETASK::Button18Click(TObject *Sender)
{

torch->Caption="Ввимкнено";
Image_torch_on->Visible=true;
Image_torch_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,1,18,1);
ClientSocket->SendLeviton_TELETASK_(0,1,TrackBar_torch->Position);

corridor->Caption="Ввимкнено";
Image_corridor_on->Visible=true;
Image_corridor_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,2,18,1);
ClientSocket->SendLeviton_TELETASK_(0,2,TrackBar_corridor->Position);

drawing_room->Caption="Ввимкнено";
Image_drawing_room_on->Visible=true;
Image_drawing_room_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,3,18,1);
ClientSocket->SendLeviton_TELETASK_(0,3,TrackBar_drawing_room->Position);

kitchen->Caption="Ввимкнено";
Image_kitchen_on->Visible=true;
Image_kitchen_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,4,18,1);
ClientSocket->SendLeviton_TELETASK_(0,4,TrackBar_kitchen->Position);

cabinet->Caption="Ввимкнено";
Image_cabinet_on->Visible=true;
Image_cabinet_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,5,18,1);
ClientSocket->SendLeviton_TELETASK_(0,5,TrackBar_cabinet->Position);

bedroom->Caption="Ввимкнено";
Image_bedroom_on->Visible=true;
Image_bedroom_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,6,18,1);
ClientSocket->SendLeviton_TELETASK_(0,6,TrackBar_bedroom->Position);

baby_room->Caption="Ввимкнено";
Image_baby_room_on->Visible=true;
Image_baby_room_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,7,18,1);
ClientSocket->SendLeviton_TELETASK_(0,7,TrackBar_baby_room->Position);

bathroom->Caption="Ввимкнено";
Image_bathroom_on->Visible=true;
Image_bathroom_off->Visible=false;
ClientSocket->Send_TELETASK_Command(0,8,18,1);
ClientSocket->SendLeviton_TELETASK_(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;
}
//-----

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

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

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

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

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

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

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

bathroom->Caption="Вимкнено";
Image_bathroom_off->Visible=true;
Image_bathroom_on->Visible=false;
ClientSocket->Send_TELETASK_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
//вмикнення та вимикнення світла випадковим чином
void __fastcall TForm_light_TELETASK::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_TELETASK::Timer1Timer(TObject *Sender)
{
int x, y;
randomize();
x=random (6)+2; //генерація випадкового номера лампи в діапазоні 2-8
ClientSocket->Send_TELETASK_Command(0,x,18,1);
y=random (6)+2;
ClientSocket->Send_TELETASK_Command(0,y,19,1);
}
//-----

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

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

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

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

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

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

```

```

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

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

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

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

ClientSocket->Send_TELETASK_StatusQuery();

st=_TELETASK_State->GetStateOnOff(0, 1);
if(st=="On") {
torch->Caption="Вімкнено";
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=_TELETASK_State->GetStateOnOff(0, 2);
if(st=="On") {
corridor->Caption="Вімкнено";
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=_TELETASK_State->GetStateOnOff(0, 3);
if(st=="On") {
drawing_room->Caption="Вімкнено";
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=_TELETASK_State->GetStateOnOff(0, 4);
if(st=="On") {
kitchen->Caption="Вимкнено";
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=_TELETASK_State->GetStateOnOff(0, 5);
if(st=="On") {
cabinet->Caption="Вимкнено";
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=_TELETASK_State->GetStateOnOff(0, 6);
if(st=="On") {
bedroom->Caption="Вимкнено";
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=_TELETASK_State->GetStateOnOff(0, 7);
if(st=="On") {
baby_room->Caption="Вимкнено";
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=_TELETASK_State->GetStateOnOff(0, 8);  
if(st=="On") {  
bathroom->Caption="Вимкнено";  
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;  
}  
}
```

КБПЗ_2023

Файл light_TELETASK.h - бібліотека для файлу light_TELETASK.cpp

```

#ifndef light_TELETASKH
#define light_TELETASKH
//-----

#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>

//-----

class TForm_light_TELETASK : public TForm
{
__published:      // IDE-managed Components
    TTrackBar *TrackBar_corridor;
    TLabel *Label11;
    TImage *Image_torch_on;
    TImage *Image_torch_off;
    TTrackBar *TrackBar_torch;
    TLabel *Label8;
    TImage *Image_corridor_on;
    TImage *Image_corridor_off;
    TTrackBar *TrackBar_drawing_room;
    TLabel *Label9;
    TImage *Image_drawing_room_on;
    TImage *Image_drawing_room_off;
    TTrackBar *TrackBar_cabinet;
    TLabel *Label12;
    TImage *Image_cabinet_on;
    TImage *Image_cabinet_off;
    TTrackBar *TrackBar_bedroom;
    TLabel *Label13;
    TImage *Image_bedroom_on;
    TImage *Image_bedroom_off;
    TTrackBar *TrackBar_baby_room;
    TLabel *Label14;
    TImage *Image_baby_room_on;
    TImage *Image_baby_room_off;
    TTrackBar *TrackBar_kitchen;
    TLabel *Label15;
    TImage *Image_kitchen_on;
    TImage *Image_kitchen_off;
    TTrackBar *TrackBar_bathroom;
    TLabel *Label16;
    TImage *Image_bathroom_on;
    TImage *Image_bathroom_off;
    TButton *Button17;
    TButton *Button18;
    TLabel *Label17;
    TLabel *torch;
    TLabel *Label19;
    TLabel *corridor;
    TLabel *Label21;
    TLabel *drawing_room;
    TLabel *Label23;
    TLabel *kitchen;
    TLabel *Label25;
    TLabel *cabinet;
    TLabel *Label27;
    TLabel *bedroom;
    TLabel *Label29;
    TLabel *baby_room;
    TLabel *Label31;

```

```

TLabel *bathroom;
TBevel *Bevel1;
TLabel *Label10;
TBevel *Bevel2;
TLabel *Label1;
TBevel *Bevel3;
TBevel *Bevel4;
TBevel *Bevel5;
TBevel *Bevel6;
TBevel *Bevel7;
TBevel *Bevel8;
TLabel *Label3;
TLabel *Label4;
TLabel *Label7;
TLabel *Label2;
TLabel *Label6;
TLabel *Label5;
TButton *torch_on;
TButton *torch_off;
TButton *corridor_on;
TButton *corridor_off;
TButton *drawing_room_on;
TButton *drawing_room_off;
TButton *kitchen_on;
TButton *kitchen_off;
TButton *cabinet_on;
TButton *cabinet_off;
TButton *bedroom_on;
TButton *bedroom_off;
TButton *baby_room_on;
TButton *baby_room_off;
TButton *bathroom_on;
TButton *bathroom_off;
TButton *Button1;
TTimer *Timer1;
TTimer *Status;
void __fastcall torch_onClick(TObject *Sender);
void __fastcall torch_offClick(TObject *Sender);
void __fastcall corridor_onClick(TObject *Sender);
void __fastcall drawing_room_onClick(TObject *Sender);
void __fastcall kitchen_onClick(TObject *Sender);
void __fastcall cabinet_onClick(TObject *Sender);
void __fastcall bedroom_onClick(TObject *Sender);
void __fastcall baby_room_onClick(TObject *Sender);
void __fastcall bathroom_onClick(TObject *Sender);
void __fastcall corridor_offClick(TObject *Sender);
void __fastcall drawing_room_offClick(TObject *Sender);
void __fastcall kitchen_offClick(TObject *Sender);
void __fastcall cabinet_offClick(TObject *Sender);
void __fastcall bedroom_offClick(TObject *Sender);
void __fastcall baby_room_offClick(TObject *Sender);
void __fastcall bathroom_offClick(TObject *Sender);
void __fastcall Button18Click(TObject *Sender);
void __fastcall Button17Click(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Timer1Timer(TObject *Sender);
void __fastcall brightnessTimer(TObject *Sender);
void __fastcall TrackBar_torchChange(TObject *Sender);
void __fastcall TrackBar_corridorChange(TObject *Sender);
void __fastcall TrackBar_drawing_roomChange(TObject *Sender);
void __fastcall TrackBar_kitchenChange(TObject *Sender);
void __fastcall TrackBar_cabinetChange(TObject *Sender);
void __fastcall TrackBar_bedroomChange(TObject *Sender);
void __fastcall TrackBar_baby_roomChange(TObject *Sender);
void __fastcall TrackBar_bathroomChange(TObject *Sender);
void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public: // User declarations
__fastcall TForm_light_TELETASK(TComponent* Owner);

```

```
};
```

```
//-----  
extern PACKAGE TForm_light_TELETASK *Form_light_TELETASK;  
//-----  
#endif
```

К6П3_2023

Файл climate_TELETASK.cpp - керування кліматом

```

#include <vcl.h>
#pragma hdrstop

#include "climate_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_climate_TELETASK *Form_climate_TELETASK;
//-----
__fastcall TForm_climate_TELETASK::TForm_climate_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm_climate_TELETASK::Button3Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,21,18,1);
}
//-----

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

```

Файл climate_TELETASK.h - бібліотека для файлу climate_TELETASK.cpp

```

#ifndef climate_TELETASKH
#define climate_TELETASKH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
//-----
class TForm_climate_TELETASK : public TForm
{
__published:      // IDE-managed Components
    TGroupBox *GroupBox2;
    TLabel *Label13;
    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: // User declarations
public: // User declarations
    __fastcall TForm_climate_TELETASK(TComponent* Owner);
};
//-----
extern PACKAGE TForm_climate_TELETASK *Form_climate_TELETASK;
//-----
#endif

```

Файл water_TELETASK.cpp - керування системою водопостачання

```

#include <vcl.h>
#pragma hdrstop

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

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

//відкриття клапану гарячої води у ванні
void __fastcall TForm_water_TELETASK::Button8Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,11,18,1);
}
//-----

//ввімкнення системи контролю протікання води на кухні
void __fastcall TForm_water_TELETASK::Button4Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,12,18,1);
}
//-----

//відкриття клапану холодної води на кухні
void __fastcall TForm_water_TELETASK::Button10Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,13,18,1);
}
//-----

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

//вимкнення системи контролю протікання води у ванній
void __fastcall TForm_water_TELETASK::Button1Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,9,19,1);
}
//-----

//закриття клапану холодної води у ванні
void __fastcall TForm_water_TELETASK::Button5Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,10,19,1);
}

```

```

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

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

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

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

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

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

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

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

```

Файл water_TELETASK.h - бібліотека для файлу water_TELETASK.cpp

```

#ifndef water_TELETASKH
#define water_TELETASKH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//-----

class TForm_water_TELETASK : public TForm
{
__published:      // IDE-managed Components
    TGroupBox *GroupBox1;
    TButton *Button1;
    TButton *Button2;
    TLabel *Label1;
    TLabel *Label_1;
    TGroupBox *GroupBox2;
    TLabel *Label3;
    TLabel *Label_4;
    TButton *Button3;
    TButton *Button4;
    TGroupBox *GroupBox3;
    TLabel *Label5;
    TLabel *Label_2;
    TButton *Button5;
    TButton *Button6;
    TGroupBox *GroupBox4;
    TLabel *Label7;
    TLabel *Label_3;
    TButton *Button7;
    TButton *Button8;
    TGroupBox *GroupBox5;
    TLabel *Label9;
    TLabel *Label_5;
    TButton *Button9;
    TButton *Button10;
    TGroupBox *GroupBox6;
    TLabel *Label11;
    TLabel *Label_6;
    TButton *Button11;
    TButton *Button12;
    TLabel *Label13;
    TLabel *Label_11;
    TLabel *Label15;
    TLabel *Label_44;
    TTimer *Status;
    void __fastcall Button2Click(TObject *Sender);
    void __fastcall Button6Click(TObject *Sender);
    void __fastcall Button8Click(TObject *Sender);
    void __fastcall Button4Click(TObject *Sender);
    void __fastcall Button10Click(TObject *Sender);
    void __fastcall Button12Click(TObject *Sender);
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button5Click(TObject *Sender);
    void __fastcall Button7Click(TObject *Sender);
    void __fastcall Button3Click(TObject *Sender);
    void __fastcall Button9Click(TObject *Sender);
    void __fastcall Button11Click(TObject *Sender);
    void __fastcall StatusTimer(TObject *Sender);

```

```
private:    // User declarations
public:    // User declarations
    __fastcall TForm_water_TELETASK(TComponent* Owner);
};
//-----
extern PACKAGE TForm_water_TELETASK *Form_water_TELETASK;
//-----

#endif
```

К6П3_2023

Файл gas_TELETASK.cpp - керування системою газопостачання

```

#include <vcl.h>
#pragma hdrstop

#include "gas_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_gas_TELETASK *Form_gas_TELETASK;
//-----
__fastcall TForm_gas_TELETASK::TForm_gas_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm_gas_TELETASK::Button2Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,15,18,1);
}
//-----

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

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

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

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

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

}

```

Файл gas_TELETASK.h - бібліотека для файлу gas_TELETASK.cpp

```

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

```

Файл tv_TELETASK.cpp - керування домашнім кінотеатром

```

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

#include "tv_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_tv_TELETASK *Form_tv_TELETASK;
//-----
__fastcall TForm_tv_TELETASK::TForm_tv_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm_tv_TELETASK::TrackBar1Change(TObject *Sender)
{
ClientSocket->SendLeviton_TELETASK_(0,32,TrackBar1->Position);
}
//-----

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

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

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

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

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

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

}
//-----

```

Файл tv_TELETASK.h - бібліотека для файлу tv_TELETASK.cpp

```

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

```

Файл phone_TELETASK.cpp - керування телефонним зв'язком

```

#include <vcl.h>
#pragma hdrstop

#include "phone_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_phone_TELETASK *Form_phone_TELETASK;
//-----
__fastcall TForm_phone_TELETASK::TForm_phone_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

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

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

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

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

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

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

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

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

ClientSocket->Send_TELETASK_StatusQuery();
}

```

```
st=_TELETASK_State->GetStateOnOff(0, 30);  
if(st=="On") Label_1->Caption="Ввімкнено"; else Label_1->Caption="Вимкнено";  
  
ClientSocket->Send_TELETASK_StatusQuery();  
st=_TELETASK_State->GetStateOnOff(0, 31);  
if(st=="On") Label_2->Caption="Ввімкнено"; else Label_2->Caption="Вимкнено";  
  
}  
//-----
```

КБПЗ_2023

Файл phone_TELETASK.h - бібліотека для файлу phone_TELETASK.cpp

```

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

```

Файл security_TELETASK.cpp - керування системою безпеки

```

#include <vcl.h>
#pragma hdrstop

#include "security_TELETASK.h"
#include "CPU_TELETASK_Socket"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm_security_TELETASK *Form_security_TELETASK;
//-----
__fastcall TForm_security_TELETASK::TForm_security_TELETASK(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm_security_TELETASK::Button14Click(TObject *Sender)
{
ClientSocket->Send_TELETASK_Command(0,17,18,1);
}
//-----

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

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

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

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

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

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

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

//визначення та виведення стану системи безпеки
void __fastcall TForm_security_TELETASK::StatusTimer(TObject *Sender)
{

```

```
ShortString st;
int n;

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

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

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

КБПЗ_2023

Файл security_TELETASK.h - бібліотека для файлу security_TELETASK.cpp

```

#ifndef security_TELETASKH
#define security_TELETASKH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Mask.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm_security_TELETASK : public TForm
{
__published:      // IDE-managed Components
    TButton *Button12;
    TGroupBox *GroupBox1;
    TLabel *Label14;
    TLabel *Label_1;
    TLabel *Label13;
    TLabel *Label_11;
    TButton *Button13;
    TButton *Button14;
    TGroupBox *GroupBox2;
    TLabel *Label6;
    TLabel *Label_3;
    TButton *Button15;
    TButton *Button16;
    TGroupBox *GroupBox3;
    TLabel *Label8;
    TLabel *Label_4;
    TButton *Button17;
    TButton *Button18;
    TGroupBox *GroupBox4;
    TLabel *Label10;
    TLabel *Label_2;
    TLabel *Label12;
    TLabel *Label_22;
    TButton *Button19;
    TButton *Button20;
    TTimer *Status;
    void __fastcall Button14Click(TObject *Sender);
    void __fastcall Button20Click(TObject *Sender);
    void __fastcall Button16Click(TObject *Sender);
    void __fastcall Button18Click(TObject *Sender);
    void __fastcall Button13Click(TObject *Sender);
    void __fastcall Button19Click(TObject *Sender);
    void __fastcall Button15Click(TObject *Sender);
    void __fastcall Button17Click(TObject *Sender);
    void __fastcall StatusTimer(TObject *Sender);
private: // User declarations
public:  // User declarations
    __fastcall TForm_security_TELETASK(TComponent* Owner);
};
extern PACKAGE TForm_security_TELETASK *Form_security_TELETASK;
#endif

```

Файл about_TELETASK.cpp – довідка про програму

```

#include <vcl.h>
#pragma hdrstop

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

void __fastcall TForm_about_TELETASK::FormCreate(TObject *Sender)
{
Memor1->Lines->Add("");
Memor1->Lines->Add("");
Memor1->Lines->Add(" МАГІСТЕРСЬКА РОБОТА");
Memor1->Lines->Add("");
Memor1->Lines->Add("на тему:");
Memor1->Lines->Add("");
Memor1->Lines->Add(" Дослідження та програмна реалізація системи керування
компонентами інтелектуального будинку за технологією TELETASK");
Memor1->Lines->Add("");
Memor1->Lines->Add("");
Memor1->Lines->Add("");
Memor1->Lines->Add("Керівник: Лисенко І.А.");
Memor1->Lines->Add("");
Memor1->Lines->Add("Розробив: студент Разумовський Максим Дмитрович");
Memor1->Lines->Add(" гр. КН-22МЗ");
Memor1->Lines->Add("");
Memor1->Lines->Add("");
Memor1->Lines->Add("м. Кропивницький 2023");
}

```

Файл about_TELETASK.h - бібліотека для файлу about_TELETASK.cpp

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