

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи обміну даними
навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-
3.0”

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Молчанов Л.К.
« ____ » _____ 2024 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О.В.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2024 року

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

Молчанову Леоніду Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0

2. Керівник роботи Коваленко Олександр Володимирович, докт. техн. наук, професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0

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

- | | |
|--|--|
| <u>1. Призначення та область використання.</u> | <u>6. Наукова новизна.</u> |
| <u>2. Перегляд аналогічних існуючих систем.</u> | <u>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</u> |
| <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки безпеки.</u> |
| <u>4. Етапи програмування системи.</u> | <u>9. Висновки.</u> |
| <u>5. Впровадження системи в промислову експлуатацію</u> | |

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

- | | |
|--|-----------------|
| <u>Наукова новизна</u> | <u>1 аркуш</u> |
| <u>Структурна схема системи</u> | <u>1 аркуш</u> |
| <u>Функціональна схема системи</u> | <u>1 аркуш</u> |
| <u>Діаграма процесів</u> | <u>1 аркуш</u> |
| <u>Блок-схема алгоритму роботи додатку</u> | <u>2 аркуша</u> |
| <u>Показники економічної ефективності</u> | <u>1 аркуш</u> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 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	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Молчанов Л.К. Дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Метою розробки є дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Об'єктом дослідження є процес обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Предметом дослідження є методи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Методи дослідження базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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

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

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

Програму розроблено в середовищі Borland C++.

Ключові слова: комп'ютерна інженерія, USB-3.0

ABSTRACT

Molchanov L.K. Research and software implementation of the data exchange system of the educational complex of micro-computers with a PC via the USB-3.0 interface. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the data exchange system of the educational complex of micro-computers with a PC via the USB-3.0 interface.

The purpose of the development is the research and software implementation of the data exchange system of the educational complex of micro-computers with a PC via the USB-3.0 interface.

The object of the research is the process of data exchange of the educational complex of micro-computers with a PC via the USB-3.0 interface.

The subject of the research is the methods of data exchange of the educational complex of micro-computers with a PC via the USB-3.0 interface.

The research methods are based on the methods of computer engineering, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of the data exchange system of the educational complex of micro-computers with a PC via the USB-3.0 interface.

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 the software are provided.

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

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

Keywords: computer engineering, USB-3.0

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.....	8
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.....	21
3.3 Розробка функціональної схеми	27
3.4 Розробка діаграми процесів.....	28
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	30
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	30
4.2 Захист розробленого програмного забезпечення.....	46
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	52
6 НАУКОВА НОВИЗНА	54

					ВКРМ-123.24.0023.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Молчанов Л.К.</i>					М	1	83
<i>Перев.</i>	<i>Коваленко О.В.</i>					ЦНТУ КІ-23М		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	55
7.1	Визначення цільової аудиторії кінцевого готового продукту	55
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	56
7.3	Вибір методу оцінки вартості ПЗ	57
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	58
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	60
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	61
7.7	Визначення ключових факторів успіху конкретного проєкту.....	63
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	65
8.1	Вступ.....	65
8.2	Характеристика умов праці програміста	66
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК.....	68
8.4	Розробка заходів з охорони праці	70
8.5	Розрахункова частина	71
9	ОСНОВНІ ВИСНОВКИ.....	75
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77

КБІП-2024

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АБГШ	–	аддитивний білий гаусів шум
ВКЗ	–	відеоконференцзв'язок
ЕОМ	–	електронно-обчислювальна машина
ІМС	–	інтегральна мікросхема
ARQ	–	автоматичний запит повторної передачі
CD-DA	–	Compact Disc Digital Audio
CIRC	–	Cross Interleaved Reed Solomon Code
EAB	–	Embedded Array Block, блок зосередженої пам'яті
ECC	–	error-correcting code, код корекції помилок
FEC	–	метод прямої корекції помилок
LDPC	–	коди Галлагера
NAK	–	негативне підтвердження

КБПЗ – 2024

ВСТУП

Актуальність теми. Мікрокомп'ютери – це один із класів пристроїв обробки інформації, який активно розвивається з 2015 року. На відміну від інших мобільних пристроїв (ноутбуків, нетбуків, ультрабуків і смартфонів), мікрокомп'ютери, як і стаціонарні настільні комп'ютери, легко спеціалізуються шляхом варіювання периферійних пристроїв і програмного забезпечення. В даний час існують мікрокомп'ютери з операційними системами Windows 10/11, Android і Linux. Як правило, наявність BIOS передбачають вибір пріоритетної використовуваної операційної системи. Аналіз варіантів використання показав, що найширший спектр додатків надає операційна система Windows 10/11, і пристрої на вищевказаній платформі були обрані для експериментів зі спеціалізації мікрокомп'ютерів для різних застосувань. З моменту випуску 29 липня 2015 року операційна система Windows 10/11 [1] перебуває у фазі активної адаптації до поліморфних пристроїв користувача (сервери, настільні комп'ютери, ноутбуки, ультрабуки, нетбуки, міні-комп'ютери, мікрокомп'ютери (microPC), мобільні телефони та інші переносні персональні пристрої) відповідно до парадигми BYOD [2]. Про це свідчать адаптація всіх елементів інтерфейсу Windows 10/11 для використання на мобільних пристроях, впровадження системи моніторингу та економії енергоспоживання апаратними та програмними компонентами, комплексне використання телеметрії для контролю роботи компонентів у порядку. для виправлення системних дефектів та клонування своїх програмних продуктів у простори інших операційних середовищ (пристрої Apple та Android).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Огляд існуючих систем обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

– Дослідження системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

– Програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Об'єктом дослідження є процес обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Предметом дослідження є методи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Методи дослідження базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

– Розроблено вітчизняний продукт обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, а також

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ - 2024

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Мікрокомп'ютери – це один із класів пристроїв обробки інформації, який активно розвивається з 2015 року. На відміну від інших мобільних пристроїв (ноутбуків, нетбуків, ультрабуків і смартфонів), мікрокомп'ютери, як і стаціонарні настільні комп'ютери, легко спеціалізуються шляхом варіювання периферійних пристроїв і програмного забезпечення. В даний час інші мікрокомп'ютери з операційними системами Windows 10/11, Android і Linux. Як правило, наявність BIOS передбачають вибір вибіркової використовуваної операційної системи. Аналіз варіантів використання показав, що найбільший спектр додатків надає операційна система Windows 10/11, і пристрої на вищевказаній платформі були обрані для експериментів зі спеціалізації мікрокомп'ютерів для різних програм. З моменту випуску 29 липня 2015 року операційна система Windows 10/11 [1] переходить у фазу активної адаптації до поліморфних пристроїв користувача (сервери, настільні комп'ютери, ноутбуки, ультрабуки, нетбуки, міні-комп'ютери, мікрокомп'ютери (microPC), мобільні телефони та інші переносні персональні пристрої) відповідно до парадигми BYOD [2]. Про це свідчать адаптація всіх елементів інтерфейсу Windows 10/11 для використання на мобільних пристроях, впровадження систем моніторингу та економічного енергоспоживання апаратними та програмними компонентами, комплексне використання телеметрії для контролю роботи компонентів у порядку. для виправлення системних дефектів та клонування своїх програмних продуктів у просторі інших операційних середовищ (пристроїв Apple та Android).

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Вибір мікроПК для експериментів з модернізації інформаційного середовища проекту зумовлений їх мобільністю за стандартом BYOD [2], на відміну від міні-комп'ютерів розміром з книжку формату А5 [7]. Деякі моделі мікроPC позиціонуються як розважальні інтернет-медіаплеєри для домашніх кінотеатрів [8], інші рекомендовані для офісних, публічних (онлайн довідкові системи), а також спеціальних програм (моніторинг систем безпеки) [4], а деякі моделі призначені для використовувати в науковій та навчальній діяльності під час читання лекцій та доповідей [3, 5]. Метою даного дослідження було визначення мінімальної апаратно-програмної конфігурації сучасних мікроПК для використання в науковій, освітній, офісній та торговій сферах діяльності.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0023.00.00.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

USB 3.0 Loopback Plugs

Програмне забезпечення призначено для усунення несправностей і тестування портів USB 3.0.

- Найшвидший спосіб перевірити, чи ваші порти USB 3.0 працюють правильно
- Діагностика, усунення несправностей і тестування під навантаженням функціональності USB вашого ПК
- Для використання з (і без) сумісного програмного забезпечення для тестування PassMark (див. нижче).

Ці тестові пристрої унікальні! Вони були спеціально розроблені для використання з нашим програмним пакетом BurnInTest Professional, BurnIntest для Linux або окремою програмою USB3Test (постачається разом із пристроєм).

Вони не вимагають зовнішнього джерела живлення та підключення до будь-якого іншого порту на ПК. Кожен штекер також має власний серійний номер, збережений у EPROM штекера, тому їх можна ідентифікувати окремо, коли підключено кілька штекерів.

Кожен плагін також містить власний ЦП і виконує власну прошивку.

Попередня перевірка перенапруги за допомогою USB SOS

Захистіть штекер USB3.0 Loopback, попередньо перевіривши несправні порти на перенапругу за допомогою тестера короткого замикання PassMark USB SOS. Цей міцний і простий тестовий штекер надасть вам візуальну індикацію напруги порту за лічені секунди, захищаючи ваш пристрій USB 3.0 Loopback від можливого пошкодження. Це особливо корисно під час тестування виробничої лінії.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Якщо ваш попередній тест USB SOS повернувся з пропуском, ви можете перейти до більш комплексного усунення несправностей за допомогою USB3.0 Loopback і супутнього програмного забезпечення.

Таблиця 2.1 – Технічні характеристики

Код товару	PM065 (раніше відомий як PMUSB03)
стандарт USB	3.0 SuperSpeed, HighSpeed і FullSpeed
Plug and play	Поступливий
Фізичний роз'єм	Стандартний 10-провідний роз'єм USB3 Micro B (гніздо)
Режими USB-передачі	Масовий та ізохронний
Напруга	4,6 В - 6,0 В. (Зовнішнє джерело живлення не потрібне)
поточний	75 мА (типова робота USB 2.0), 90 мА (типова робота USB 3.0)
Тактова частота	19,2 МГц
Пам'ять пристрою	512 Кб
Справа	Ударостійкий ABS-пластик
Розмір	85 мм x 45 мм x 12 мм (3,4 x 1,8 x 0,5 дюйма)
вага	40 г (1,4 унції)
Індикатори	Помаранчевий світлодіод = Живлення від шини USB Зелений світлодіод = Передача даних Жовтий світлодіод = Отримання даних Червоний світлодіод = Індикатор помилки передачі шини USB
Стандарти EMC	AS/NZS 3548:1995, EC.
Температура зберігання	від -30 °C до 80 °C
Температура використання	від 0 °C до 70°C
Максимальна швидкість	5 Гбіт/сек

З якими операційними системами сумісні ці модулі?

Наразі драйвери доступні лише для Windows 7, 8, 10, 11 і 2016, 2019, 2022 Server (у 32- та 64-розрядних версіях). Також є підтримка Windows з процесорами x86 і ARM.

Підтримується Linux із версіями ядра 2.6.31 або новішими, якщо використовується з BurnInTest для Linux версії 3.1 (або новішої). також потрібна libusb 1.0. (Це стандартна бібліотека Linux, яка постачається майже з усіма дистрибутивами Linux).

Slax Linux (7.0.8) не підтримується, оскільки, схоже, існує можлива помилка драйвера пристрою, яка заважає роз'єму USB3 увійти в режим зворотного зв'язку належним чином, інші перевірені живі системи (Kubuntu 12.04, Fedora 18, Porteus 3.0.1) не здаються бути ураженим тією ж помилкою.

Чи потрібен мені драйвер пристрою для використання цих розеток?

Для Windows потрібен драйвер пристрою. Під час першого використання розетки Windows запитає драйвер пристрою. Для Linux не потрібен спеціальний драйвер пристрою.

Чи сумісні драйвери пристрою з усіма версіями BurnInTest і USB3Test?

Останні версії BurnInTest і USB3Test працюють із будь-якою версією драйвера USB 3.0 Loopback plug device, проте рекомендується використовувати останню версію.

Чи з'являється роз'єм із петлею PassMark USB 3.0 у диспетчері пристроїв як новий USB-пристрій?

Так, ви можете бачити пристрої в диспетчері пристроїв Windows. Вони з'являються з міткою «PassMark USB3.0 Loopback Plug».

USB Power Delivery Tester

Для усунення несправностей портів USB і зарядних пристроїв USB

– Перевірте потужність портів USB, портів Thunderbolt, зарядних пристроїв USB і блоків живлення USB.

– Перевірте, чи USB-хост здатний забезпечити максимальну вказану потужність без збоїв.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Визначайте потужність зарядного пристрою USB (wall-wart), який, у свою чергу, контролює швидкість заряджання мобільних телефонів та інших акумуляторних пристроїв, підключених через USB.

– Переконайтеся, що рівні напруги залишаються в межах специфікації під високим навантаженням.

– Виміряйте точне енергоспоживання пристроїв USB.

– Використовуйте його разом із штекером USB 3.0 Loopback Plug, щоб одночасно перевірити швидкість зв'язку, цілісність даних і подачу живлення USB-портів.

– Підтримка QC4+ і PPS доступна в моделі PM125.

Техніка безпеки

Не торкайтеся і не накривайте радіатор під час роботи, оскільки він може нагрітися до високої температури, що може призвести до опіків або виходу з ладу USB PD Tester.

Не блокуйте роботу вентиляторів і забезпечте хороший потік повітря. Залиште принаймні 10 см повітряного зазору навколо задньої частини вентиляторів. Також переконайтеся, що під пристроєм є потік повітря.

Використовуйте тестер USB PD лише з пристроями, які стверджують, що відповідають стандартам USB. Катастрофічний збій тестованого пристрою може призвести до короткого замикання живлення від системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 USB-кабелю. Це може як зруйнувати USB-тестер PD, так і створити серйозну небезпеку ураження електричним струмом.

USB PD Tester не є споживчим пристроєм. Його розроблено для використання кваліфікованими інженерами-електриками.

Завжди нехай хтось стежить за тестуванням на випадок, якщо тестований пристрій раптом вийде з ладу. Не починайте тест і йдіть геть.

Відомо, що деякі блоки живлення USB не повністю відповідають стандартам USB. Деякі також не відповідають відповідним електричним стандартам у країнах, де вони продаються. Несправні або погано розроблені пристрої можуть бути небезпечними! Тестер USB PD може виявити дефекти

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

конструкції та виробництва пристрою під час високого навантаження. Результатом може стати катастрофічний вихід тестованого пристрою з ладу. Що, у свою чергу, може призвести до пожежі, розплавлення тестованого пристрою, задимлення, короткого замикання та навіть руйнування самого USB-тестера PD.

Не споживайте більше струму від пристрою, ніж він стверджує, що підтримує. У результаті порт може бути пошкоджено, або захист від перевантаження може призвести до вимкнення порту.

USB S.O.S. (Short circuit - Over voltage - Sensor)

Тестер короткого замикання для портів USB.

– Найшвидший і найпростіший спосіб перевірити напругу та коротке замикання в портах USB і зарядних пристроях USB.

– Випробуйте підвищену або знижену напругу на VBUS.

– Перевірте, чи немає короткого замикання між VBUS і будь-якою з ліній даних.

– Виявлення недійсної напруги на лініях передачі даних.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

На заміну старого розробленого набору елементів управління у Builder C++ інтегрована бібліотека візуальних компонентів VCL, представлених на палітрі компонентів. Після переносу на форму методом перетягування (drag-and-drop) компоненти відразу становляться діючими об'єктами вашої програми.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Окрім типізованих інтерфейсних елементів Windows (кнопки, смуги прокручування, редагуємі текстові області, прості та комбіновані списки, та інше) у бібліотеку включені елементи підтримки діалогових вікон, обслуговування баз даних та багато іншого. Можливо не тільки модифікувати поведінку існуючих компонентів, але і будувати нові.

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

Builder C++ підтримує зв'язок з різними базами даних 3-х видів: dBASE і Paradox; Sybase, Oracle, InterBase і Informix; Excel, Access, FoxPro і 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.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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 випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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;

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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.

КБПЗ – 2024

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

USB 3.0 – це стандарт передачі даних універсальної послідовної шини (USB), який зазвичай використовується для комп'ютерної периферії. Здебільшого USB 3.0 зустрічається на роз'ємах типу USB-A або USB-C. USB 3.0 зазнав кількох змін і підтримує швидкість від 5 до 20 Гбіт/с. USB 3.0 також відомий як SuperSpeed USB.

Найпоширеніший сьогодні порт для периферійних пристроїв, USB 3.0, часто використовується для підключення зовнішніх високошвидкісних пристроїв, таких як флеш-накопичувачі, мережеві адаптери та камери високої роздільної здатності. Оскільки USB 3.0 зворотно сумісний із пристроями USB 2.0 і USB 1.1, більшість периферійних пристроїв, таких як миші, клавіатури та принтери, можна підключати до них.

Thunderbolt і USB4 (він же USB 4.0) є новішими стандартами, які забезпечують вищу швидкість і більше можливостей, ніж USB 3.0. Незважаючи на це, USB 3.0 залишається популярним роз'ємом завдяки простоті впровадження та низькій вартості для виробників, а також широкій сумісності.

Сучасні комп'ютери часто містять порти USB 3.0 як роз'єми USB A та USB C. USB 3.0 все ще забезпечує достатню швидкість майже для всіх споживчих цілей, за винятком надзвичайно швидкої зовнішньої передачі даних на основі флеш-пам'яті.

USB 3.0 зазвичай використовується для підключення зовнішніх високошвидкісних пристроїв, таких як флеш-накопичувачі.

Яка історія та покоління USB 3.0?

USB 3.0 вперше було представлено в листопаді 2008 року, а споживчі пристрої стали доступними в 2010 році. У галузі розроблено його, щоб збільшити

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

максимальну пропускну здатність стандарту USB, щоб конкурувати з іншими роз'ємами, такими як firewire , зберігаючи при цьому зворотну сумісність з існуючими пристроями USB 2.0. USB 3.0 також додає повнодуплексний зв'язок.

Перша версія була USB 3.0 і підтримувала швидкість до 5 Гбіт/с. Новіші версії також називатимуть це USB 3.1 Gen 1 і USB 3.2 Gen 1x1.

У січні 2013 року стандарт було оновлено до USB 3.1. Це запровадило новий режим під назвою SuperSpeed+ або USB 3.1 Gen 2. Він міг працювати зі швидкістю 10 Гбіт/с. Він також має краще кодування з меншими витратами на кодування, що призводить до вищої ефективної швидкості передачі даних.

Стандарт USB 3.2 був представлений у липні 2017 року. Стандарт USB 3.2 включає новий режим передачі 20 Гбіт/с, який працює з роз'ємами USB-C. Він відомий як SuperSpeed+ або USB Gen 2x2. Стандарт 3.2 також перейменував старіші швидкості передачі.

Через дещо складне найменування нові пристрої USB 3.0 повинні містити максимальну підтримувану швидкість передачі в логотипі пристрою. На щастя, усі стандарти швидкості передачі даних USB зворотно сумісні, тобто майже будь-який пристрій можна використовувати з будь-яким штекером, і він автоматично повернеться до найкращої взаємно підтримуваної швидкості.

Стандарт USB4 був представлений у серпні 2019 року. Він забезпечує кращу швидкість, потужність і відеовихід, зберігаючи зворотну сумісність із USB 3.0.

Які є різні типи роз'ємів USB 3.0?

Для підтримки збільшеної швидкості USB 3.x потребує п'яти контактів більше, ніж USB 2.0. У багатьох випадках новий роз'єм був розроблений для підтримки фізичної сумісності зі старими роз'ємами.

Роз'єм **USB A** – це прямокутний роз'єм, який найчастіше асоціюється з USB. Додаткові штифти були додані до внутрішніх частин вилки, в результаті чого вилка виглядає майже так само. Щоб відрізнити USB 3 від старих стандартів, пластикові вставки мають бути синього кольору, зокрема Pantone

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

300С. Однак синій колір є лише рекомендацією, тому багато виробників пристроїв уникають цього кольору з естетичних міркувань.

Роз'єм **USB B** – це квадратний роз'єм, який найчастіше можна побачити на зовнішніх клієнтських пристроях, таких як принтери або деякі зовнішні жорсткі диски. Він функціонально замінений на USB C.

Роз'єм **Micro USB B** – це плоский і маленький роз'єм, який можна побачити на таких пристроях, як зовнішні жорсткі диски, цифрові камери та деякі смартфони. Він функціонально замінений на USB C.

Роз'єм **USB C** – це овальний штекер, який можна побачити на головних комп'ютерах і клієнтських пристроях. Він підтримує найвищу швидкість передачі даних і додаткові альтернативні режими, такі як Power Delivery або DisplayPort Out.

Для найкращої швидкості кабелі також повинні підтримувати USB 3.0. Наприклад, деякі зарядні кабелі USB C можуть бути розроблені для високошвидкісного живлення, але вони не підтримуватимуть USB 3 SuperSpeed і передаватимуть дані лише на швидкості USB 2.

Ви можете легко визначити, чи є у вас порт USB 3, якщо він синій або має логотип, що складається з літер «SS» із символом тризуба.

Що таке додаткові протоколи USB 3.0?

Роз'єми USB C містять додаткові контакти, які розширюють можливості одного штекера. Їх можна поєднувати і використовувати одночасно. Це дозволяє док-станції мати USB-концентратор і відеовихід під час заряджання ноутбука за допомогою одного кабелю. Проте слід бути обережним, оскільки не всі порти підтримують усі ці режими. Зверніться до документації виробника щодо сумісності пристрою.

Power Delivery забезпечує підтримку двонаправленої зарядки. Зазвичай він може досягати 100 Вт потужності, а нові стандарти підтримують навіть більше.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Альтернативний режим DisplayPort – це вихід цифрового відео з підтримкою кількох дисплеїв, послідовного підключення з роздільною здатністю до 8k.

Альтернативний режим HDMI – це цифровий відеовихід. Хоча існує стандарт для виходу альтернативного режиму HDMI для USB C, пристрої часто використовують DisplayPort у роз'ємі HDMI.

Mobile High-Definition link – це старіший цифровий відеостандарт для виведення HDMI через USB. Зараз використовується рідко.

Thunderbolt 3 – це ліцензована технологія Intel, яка включає в себе USB 3, Power Delivery, DisplayPort, експрес-з'єднання периферійних компонентів і можливість передачі даних 40 Гбіт/с через один порт.

Акcesуари для аудіоадаптера передають аналоговий звук через контакти USB C. Він може надсилати стереовихід і мікрофон. Він використовувався на деяких смартфонах, але зараз рідко зустрічається.

У фірми Microchip є відповідні для цієї справи мікроконтролери PIC18F2455/2550/4455/4550. Вони мають в своєму складі модуль USB. Я в цій роботі орієнтуватимуся на мікроконтролер PIC18F4550

Отже, потрібно написати програму для мікроконтролера PIC18F4550. Таку програму щоб при підключенні цього мікроконтролера до комп'ютера відбувалося створення віртуального COM-порту, через який можна передавати і приймати дані. Задача не з адже легких

Опис USB і програми вирішино почати не з утомливої теорії, а з найцікавішого, тобто з практики. В цій дипломній роботі хочеться щоб ви відчули як це, коли твоя робота при підключенні до USB розпізнається комп'ютером як пристрій готовий до роботи. Для цього потрібно зібрати наступну схему.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

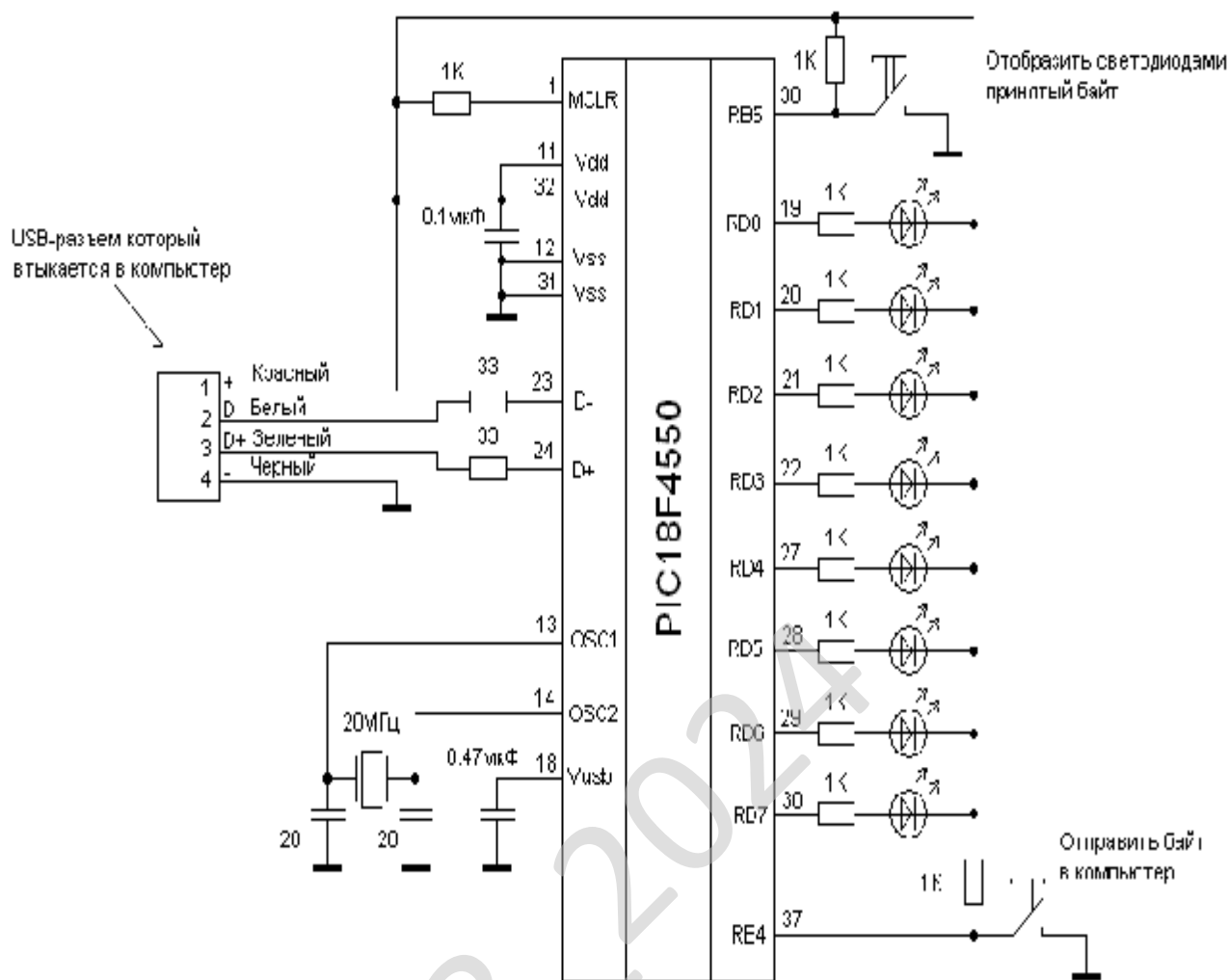


Рисунок 3.1 – Підключення мікропроцесора PIC18F4550 до ПК через порт USB

3.2 Розробка структурної схеми

Рекомендовані карти хост-контролера USB 3.0

Ми протестували такі набори мікросхем USB 3.0:

- Серії Intel 7, 8 і 9.
- VIA.
- Renesas (раніше NEC, той самий чіп).
- ETRON.
- Texas Instruments.

Примітка щодо купівлі хост-контролерів від Amazon: оскільки всі хост-контролери USB 3.0 реалізують загальну специфікацію (XHCI), безіменні постачальники USB 3.0 можуть легко вибрати, який хост-чіп вони використовують залежно від наявності чи вартості. Зазвичай це помічають люди у відгуках.

Картка, яку ми посилаємо, спеціально рекламується з використанням контролера Renesas, тому ми вирішили її протестувати. Однак є принаймні один огляд, який стверджує, що їх карта постачалася з чіпсетом VIA. Будь ласка, перевірте його, коли ви його отримаєте. У Windows набір мікросхем VIA потребує оновлення драйверів для належної роботи.

Попереджувальне повідомлення: виявлено можливу проблему USB-контролера

У Windows програмне забезпечення може створити таке попередження, якщо виявить можливу несумісність USB-контролера.

Хост-контролери ASMedia та VIA

Дві відомі комбінації драйверів хост-контролера USB із проблемами (хост-контролери ASMedia та хост-контролери VIA) мають відомі робочі конфігурації драйверів.

Пошук інформації про хост-контролер USB 3.0:

1. Усунення несправностей підключення.
2. Знайшовши інформацію про хост-контролер USB 3.0, перевірте, чи є ваш пристрій Saleae Logic у списку диспетчера пристроїв.
3. Відкрийте програмне забезпечення Logic 2 і перевірте, чи підключається пристрій. Якщо ні, перевірте, чи пристрій не зник із диспетчера пристроїв.
4. Перезавантажте комп'ютер, щоб перевірити, чи проблема з підключенням не зникає.
5. Якщо проблема з підключенням не зникає навіть після перезавантаження комп'ютера, можливо, знадобиться оновлення драйвера.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

6. Якщо для вашого комп'ютера є доступний драйвер, є хороші шанси, що Windows Update його матиме. Щоб перевірити, відкрийте Windows Update, натисніть «Переглянути додаткові оновлення» та розгорніть «Оновлення драйверів».

Якщо ваш драйвер не відображається в списку «Оновлення драйверів» у Windows Update, це означає, що в Windows Update немає доступного оновлення. У цьому випадку не слід намагатися примусово інсталювати один із їхніх пакетів оновлення.

Останні драйвери хост-контролера USB 3.0

Windows 10/11 і Windows 8.1/8

Windows 10/11 і Windows 8.1/8 використовують новий драйвер USB 3.0, створений Microsoft, який підтримує всі хост-контролери USB 3.0. Цей драйвер працює надзвичайно добре, і його рекомендовано використовувати для всіх хост-контролерів USB 3.0, за винятком наступних хост-контролерів

- Intel USB 3.1 eXtensible Host Controller.

- Ми отримали повідомлення про деякі проблеми з листопадовою версією розширюваного хост-контролера Intel USB 3.1. Будь ласка, перевірте Windows Update, чи повідомляє про будь-які додаткові оновлення, доступні для вашого хост-контролера USB.

- Хост-контролер ASMedia USB 3.0 або 3.1.

- Ми рекомендуємо завантажити останню доступну версію від постачальника USB-карти чи материнської плати.

- Примітка: усі пристрої ASMedia (3.0 або 3.1) використовують той самий драйвер.

- Через USB хост-контролер.

- Посилання для завантаження драйвера.

- Лише драйвери, старіші за версію 4.90A, мають проблеми.

Для інших хост-контролерів немає необхідності виконувати будь-яке оновлення драйверів, якщо ви використовуєте одну з цих версій Windows.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

OS X

Драйвери хост-контролера USB 3.0 є частиною операційної системи OS X і не оновлюються окремо. Оновлення різних частин стека USB часто входять до оновлень Apple, тому настійно рекомендуємо постійно оновлювати свою ОС і використовувати останню версію OS X.

Примітка для OSX Bootcamp (Windows)

Якщо ви використовуєте Windows на обладнанні Apple, обов'язково оновіть програмне забезпечення Bootcamp до останньої версії. Зокрема, переконайтеся, що ви використовуєте Bootcamp 5 або новішу версію. Якщо ви використовуєте Windows 7, після встановлення Bootcamp 5+ переконайтеся, що ви можете знайти хост-контролер Intel USB 3.0 у списку диспетчера пристроїв у розділі «Контролери універсальної послідовної шини». Якщо ви не можете знайти цей пристрій, можливо, виникла проблема під час встановлення драйверів Bootcamp. У такому випадку ви можете спробувати вручну встановити драйвери USB 3.0, завантаживши zip-файл bootcamp від Apple, розпакувавши його та знайшовши інсталятор драйвера Intel USB 3.0 нижче:

BootCamp5.1.5621/BootCamp/Drivers/Intel/IntelxHCISetup.exe

Запустіть цю програму та перезавантажте комп'ютер, щоб інсталювати драйвери хост-контролера Intel USB 3.0.

Linux

У Linux усі хост-контролери USB 3.0 використовують модуль ядра xhci_hcd, який неможливо оновити окремо від ядра Linux. Загалом ми рекомендуємо використовувати ядро 3.4 або новішу (Ubuntu 14.04.1 або новішу). Хоча продукти працюють на старіших ядрах, ми офіційно не підтримуємо такі варіанти використання.

Якщо у вас виникли проблеми з пристроями, які підключаються чи записують дані в Linux, обов'язково зв'яжіться зі службою підтримки та надайте інформацію про версію ядра та встановлені хост-контролери. Зокрема, надайте результати:

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- uname -r
- lspci -k
- lsusb

Непідтримувані операційні системи

Windows 7 і старіші версії

Підтримка Microsoft Windows 7 припинилася 14 січня 2020 року.

Найпоширеніші проблеми з Logic Pro 8 і Logic Pro 16 у Windows 7 можна вирішити, просто оновивши драйвери хост-контролера USB 3.0 вручну.

Ці посилання на драйвери слід використовувати, лише якщо ви використовуєте Windows 7. Не всі драйвери доступні безпосередньо від постачальника хост-контролера, а натомість розповсюджуються платою PCI Express або постачальником материнської плати, яка інтегрувала цей хост-контролер. У більшості випадків добре використовувати драйвер для завантаження з веб-сайту, відмінного від того, на якому була продана карта, за умови, що він використовує той самий хост-контролер. Також можна завантажити драйвер безпосередньо з компанії, де ви придбали картку. У такому випадку переконайтеся, що версія драйвера така сама або новіша, ніж указана нижче. Ми рекомендуємо вам використовувати драйвери, доступні безпосередньо від виробника хост-контролера.

- Хост: Renesas/NEC Джерело: від постачальника картки чи материнської плати Версія: 2.1.36.0 Посилання: https://www.rosewill.com/rosewill-rc-505-2-port-usb-3-0-pci-express-card.html#product_tabs_Downloads

- Хост: Intel серії 7 Джерело: Direct (Intel) Версія: 1.0.10.255 Посилання: <https://downloadcenter.intel.com/download/21129/USB-3-0-Driver-Intel-USB-3-0-eXtensible-Host-Controller-Driver-for-Intel-7-Series-C216-Chipset-Family>

- Хост: Intel серії 8, 9, 100 Джерело: Direct (Intel) Версія: 4.0.0.36 Посилання: <https://downloadcenter.intel.com/download/22824/USB-3-0-Driver-Intel-USB-3-0-eXtensible-Host-Controller-Driver-for-Intel-8-9-100-Series-i-C220-C610-Chipset-Family>

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Хост: ASMedia 3.0 або 3.1 Джерело: від постачальника картки чи материнської плати Версія: 1.16.23.0 Посилання: [https://www.asus.com/Motherboard-](https://www.asus.com/Motherboard-Accessories/USB_31_TYPEA_CARD/HelpDesk_Download/)

[Accessories/USB_31_TYPEA_CARD/HelpDesk_Download/](https://www.asus.com/Motherboard-Accessories/USB_31_TYPEA_CARD/HelpDesk_Download/) Примітка: усі пристрої ASMedia (3.0 і 3.1) використовують той самий водій

- Хост: ETRON Джерело: Direct (Etron) Версія: 1.0.0.118 Посилання: http://www.etrone.com/en/products/u3hc_detail.php?Product_ID=6 Примітка. Натисніть посилання в навігаційній панелі для «Завантажити драйвер». Використовується для всіх моделей/версій

- Хост: Texas Instruments Джерело: Direct (Ti) Версія: 1.16.5 Посилання: <http://www.ti.com/product/TUSB7340/toolssoftware> Примітка: драйвер TUSB73x0 xHCI для WinXP/Vista/7 (Rev. J)

- Хост: VIA Джерело: Direct (Via) Версія: 4.90 А (6.1.7600.4903) Посилання: <http://www.via-labs.com/driver.php> Примітка: Драйвер WHQL для VL800/801 & 805/806 USB 3.0 Host Контролер. Сумісний з Windows XP/Vista/7/8 32-bit та 64-bit.

- Хост: Fresco Logic Джерело: Direct (Fresco Logic) Версія: 3.6.8.0 Посилання: <http://www.frescologic.com/support.php> Примітка: Хост-драйвер USB3.0 HostDriver_V3.6.8.0

3.3 Розробка функціональної схеми

Функціональна схема – схема, що роз'яснює певні процеси, що відбуваються у певних функціональних ланцюгах виробу (устаткування) чи у виробі (устаткуванні) в цілому.

В схемі показується зв'язок між мікропроцесором , аналоговою частиною, шиною керування , дисплеєм тощо.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

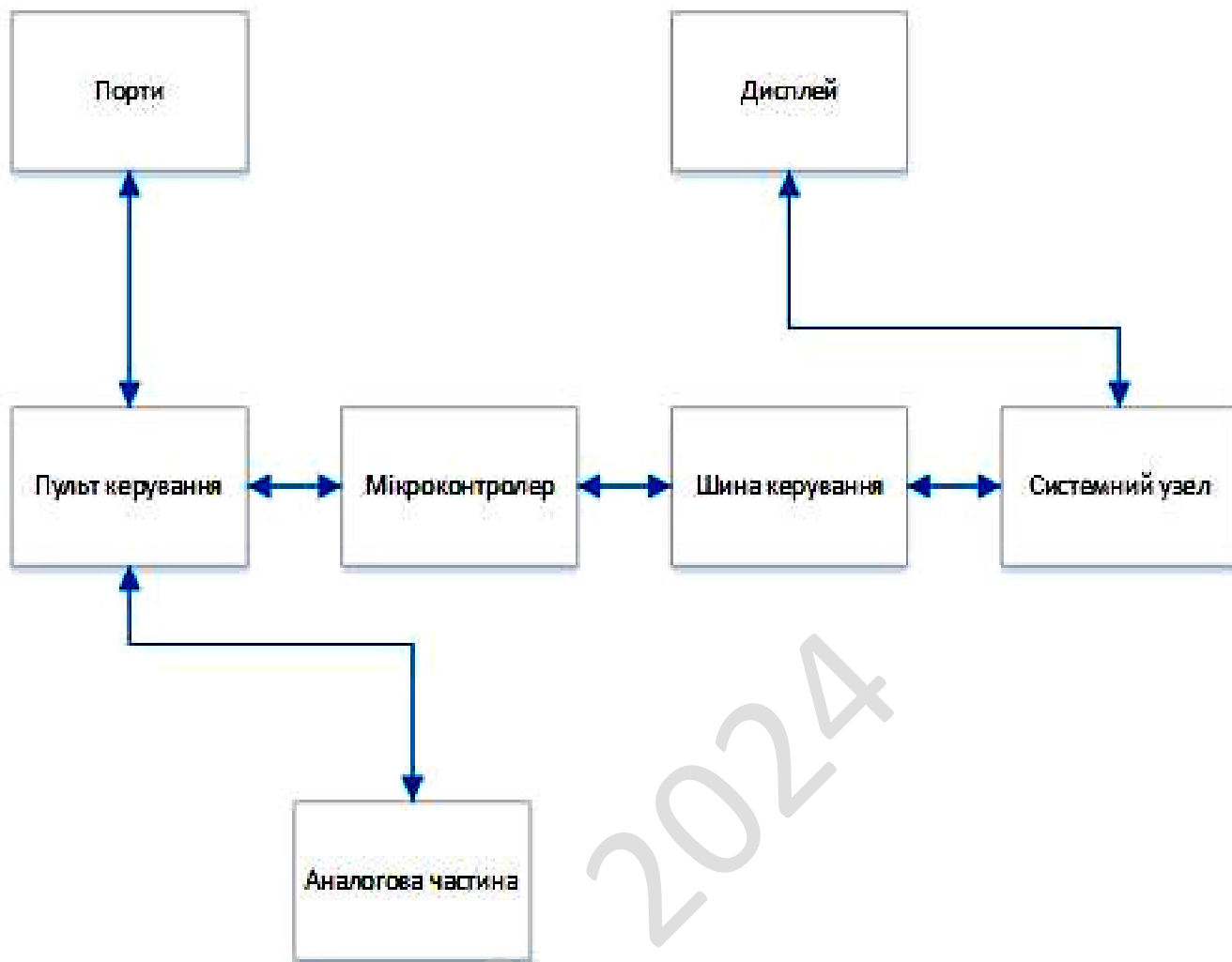


Рисунок 3.3 – Функціональна схема

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.6. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

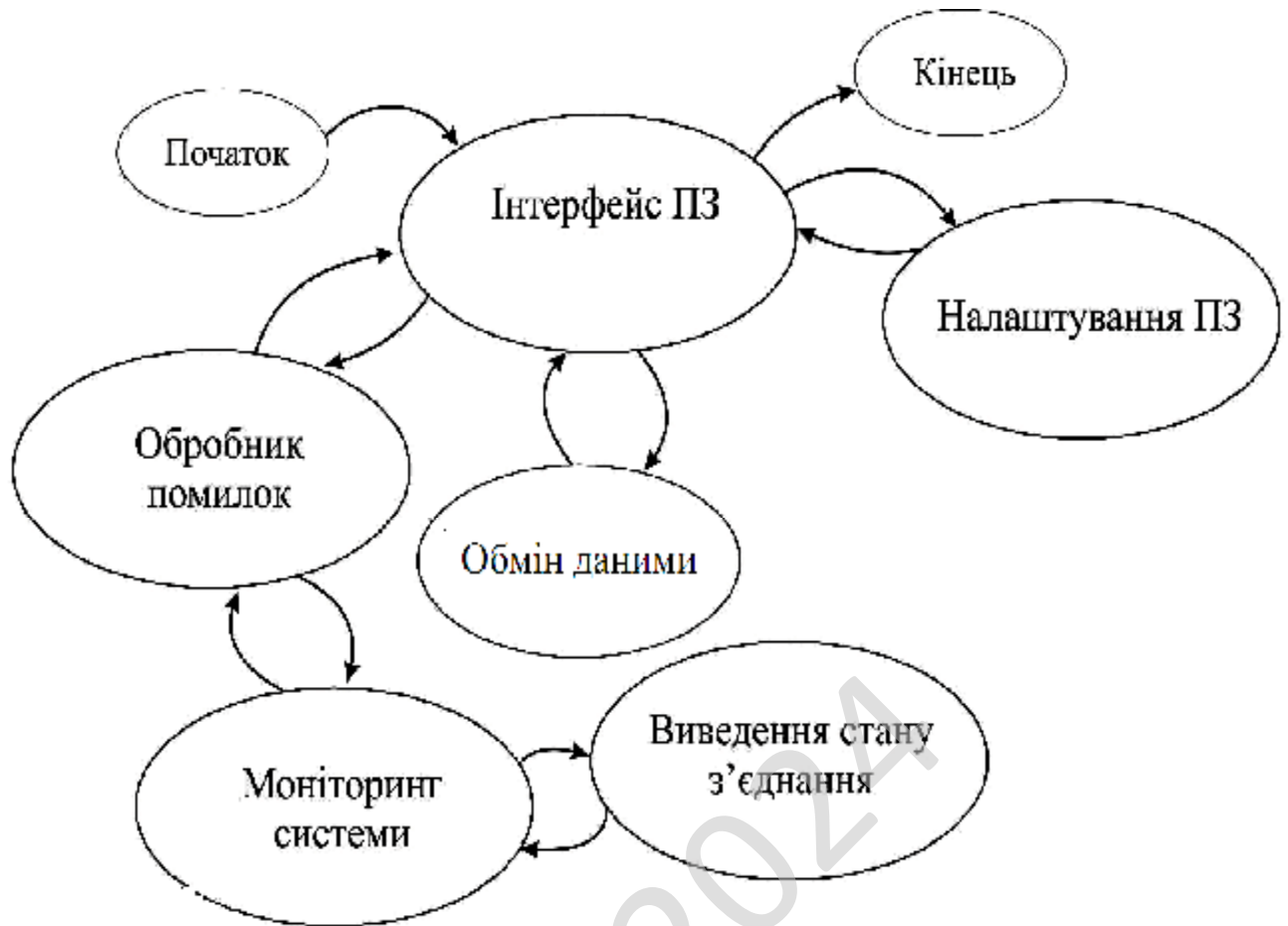


Рисунок 3.4 – Діаграма процесів

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.

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.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Для функціонування даної системи, а саме отримання та передачі інформації з мікроконтролера на персональний комп'ютер, потрібно написати драйвер на зв'язок мікроконтролера з USB та на персональний комп'ютер .

Розроблюємо драйвер на зв'язок стендом з інтерфейсом USB 3.0 .

Необхідний софт для роботи:

– AVR Studio 4 – середовище розробки від Atmel, для написання прошивки МК. WinAVR – open source набір розробника для AVR, тут нас цікавить GNU GCC компілятор, автоматом інтегрується в студію. Все що стосується мови Cі на EasyElectronics.ru

– V-USB – програмна реалізація low-speed USB 1.1 для мікроконтролерів AVR.

– Borland C ++ Builder 6 Enterprise Edition + UPDATE 4 – середовище швидкої розробки додатків на C ++ під ОС Windows

АТmega8 підключена класично з 5-ти вольтовим харчуванням (прикладі інших схем можна подивитися в каталозі circuits архіву V-USB). У USB рівень сигналу на лініях D + і D– становить 3.3V, а схему підключили до 5V, тому для узгодження встановлені стабілітрони D1 і D2, які зменшують сигнал з мікроконтролера на гасять резисторах R3, R4 до рівня необхідного за стандартом. Для визначення версії протоколу в схемі задіяний дільник напруги на резисторах R1 і R2 який забезпечує 3.4V для D– на холостому ході. Якщо замість резистора R1 2,2K поставити R1 номіналом 1,5K (або, що ще гірше, чи не встановити R2 зовсім – як в оригінальному circuit від V-USB), то замість 3,4V ми отримаємо 3.7V, що в свою чергу відкриє стабілітрон D1 який зменшить її до напруги його

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

відкривання (3.4 – 3.5V). І в результаті ми отримуємо перекис по струму в режимі холостого ходу (D2 не навантажений зовсім, а по D1 вже тече струм) що буде позначатися на довгих проводах USB. Опору R3 і R4 розраховуються по струму і для даної схеми рівні 68 Ом.

Тактова частота для мікроконтролера

Асемблерні частини V-USB написані з підтримкою тільки таких частот: 12 MHz, 12.8 MHz, 15 MHz, 16 MHz, 16.5 MHz, 18 MHz і 20 MHz. Інші не підтримуються. Частоту вручну ніде прописувати не потрібно! Її необхідно визначити в налаштуваннях проекту AVR Studio: Project -> Configuration Options - > вкладка General -> поле Frequency (в Герцах, для 12MHz додаємо приставку мега = 12000000) інакше отримуємо купу помилок і не зможемо правильно скомпілювати програму (студія створює define F_CPU для компілятора, який доступний всім проектом, V-USB якраз використовує його).

12 MHz: традиційна частота для V-USB, тому що це мінімальна частота при якій можливе емулювати всі необхідні таймінги специфікації USB.

15 MHz: схожа до 12 MHz, але місцями вставлені NOP-и. З іншого боку велика частота дозволяє частіше використовувати цикли, які роблять кінцевий код прошивки трохи меншим у порівнянні з 12 MHz.

16 MHz: підтримка цієї частоти була спеціально додана для користувачів Arduino та інших готових плат які має штатний кварц на 16 MHz. Також корисно якщо необхідно підвищити частоту в цілях продуктивності. Оскільки 16 MHz не ділиться без остачі на USB low speed bit clock 1.5 MHz, то асемблерна реалізація для даної частоти написана з деякими хитрощами, застосовуються цикли для уповільнення.

12.8 MHz і 16.5 MHz: ці частоти розраховані для тактирования від вбудованого RC генератора з точністю 1%. Генератор прийдеться калібрувати на ходу за допомогою USB frame clock. Кому цікаво працювати без кварцу можна розкурювати такий приклад – EasyLogger на ATTiny45.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

хоче відправити дані пристрою він дає команду `HID_SET_REPORT`, при цьому V-USB всередині мікроконтролера викликає функцію `usbFunctionWrite ()`. Аналогічно коли хост хоче прочитати дані з пристрою, він дає команду `HID_GET_REPORT`, при цьому V-USB викличе функцію `usbFunctionRead ()`. USB лінія даних D + підключена до переривання `INT0` (воно з найвищим пріоритетом). Коротше коли відбуватиметься обмін даними по USB, мікроконтролер постійно буде йти на обробку `INT0`, на якому висить V-USB. І тільки потім управління віддадуть основній програмі. Якщо потрібно використовувати свої переривання (бажано швидкі і компактні), то відразу ж ставимо глобальний прапор переривань і командою `sei ()`, щоб при запитах від хоста могло спрацювати `INT0` для правильної роботи V-USB.

Функція `usbPoll ()`, яку необхідно викликати не рідше ніж 50 ms – повідомляє хосту, що наш пристрій на шині USB ще живе і чекає свого часу готове до роботи. Якщо це не дотримуватися, то вінда напише «Підключено невідомий пристрій...». 50 ms – це USB timeout for accepting a Setup message (це спеціальні команди хоста, у нашому випадку – `HID_SET_REPORT` і `HID_GET_REPORT`). Setup повідомленнями займається функція `usbFunctionSetup ()`, власне тут обробляються керуючі команди USB і далі вже запускається `usbFunctionRead ()` або `usbFunctionWrite ()` в даному Hid прикладі. Ось тут добре описано пакети при передачі даних по USB.

USB HID report descriptor

Простими словами це масив констант зашитий у Flash пам'ять пристрою для опису структури пакетів даних (Hid репортів). У ньому міститься така інформація: скільки пакетів підтримує пристрій, яка їх довжина, призначення кожного байта і біта в пакеті. Таким чином при підключенні до комп'ютера пристрій за допомогою такого дескриптора повідомляє свої параметри, а ОС комп'ютера вже буде знати як правильно спілкуватися з цим пристроєм (наприклад які біти відповідають за натискання певних кнопок джойстика). У нас буде «HID сумісний пристрій» для передачі довільних даних фіксованого розміру

зберігається структура. В коментаріях Vga – «оголосити структуру і привести покажчик до неї – цілком типовий метод, коли треба працювати з блоком даних у пам'яті, який може являти собою ту чи іншу структуру». Ілюстрація пояснень прийому / передачі структури по частинах:

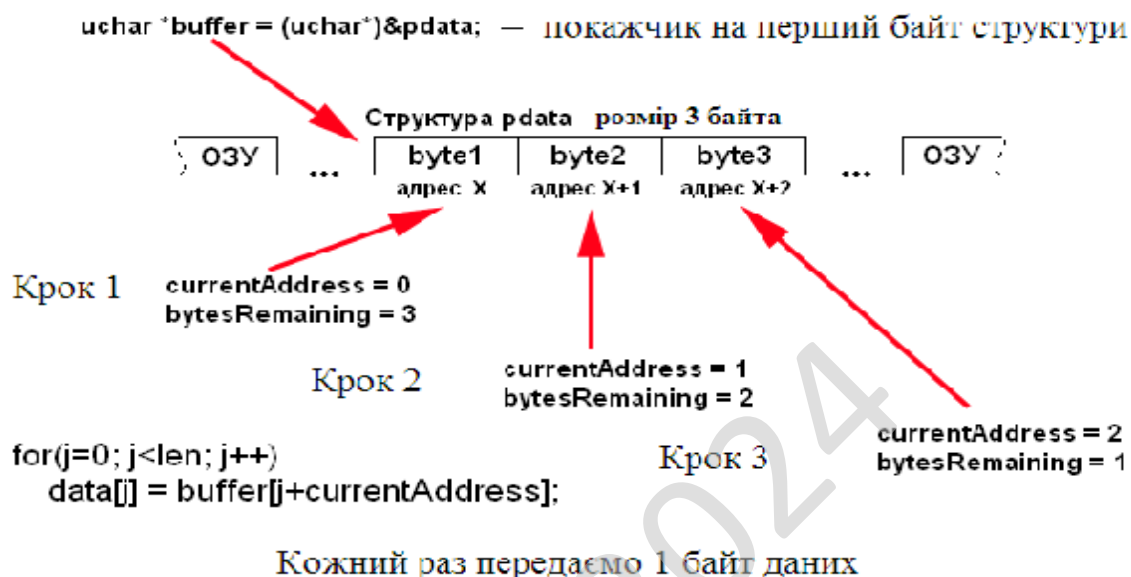


Рисунок 4.4 – Схема передачі даних

Розібравшись з усім вищесказаним, приступимо до написання програми для мікроконтролера (`main.c`)

```
#include <avr / io.h>
#include <avr / interrupt.h>
#include <util / delay.h>
#include <avr / pgmspace.h> / * потрібно для usbdrv.h * /
#include "usbdrv.h"
struct dataexchange_t // Опис структури для передачі даних
{
    uchar b1; // Вирішино для прикладу написати структуру на 3 байта.
    uchar b2; // На кожен байт підчепити ногу з PORTB. Звичайно це
    uchar b3; // Не раціонально (всього то 3 біти потрібно).
}; // Але з метою демонстрації в самий раз.
// Для наочності прикрутити по світодіоду і споглядати
struct dataexchange_t pdata = {0, 0, 0};
PROGMEM char usbHidReportDescriptor [22] = { // USB report descriptor //
Дескриптор описує структуру пакета даних для обміну
```

```

0x06, 0x00, 0xff, // USAGE_PAGE (Generic Desktop)
0x09, 0x01, // USAGE (Vendor Usage 1)
0xa1, 0x01, // COLLECTION (Application)
0x15, 0x00, // LOGICAL_MINIMUM (0) // min. значення для даних
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255) // max. значення для даних,
255 тут не випадково, а щоб укластися в 1 байт
0x75, 0x08, // REPORT_SIZE (8) // інформація передається порціями, це
розмір одного "репорт" 8 біт
0x95, sizeof (struct dataexchange_t), // REPORT_COUNT // кількість
порцій (у нашому прикладі = 3, описана вище структура передається за три репорт)
0x09, 0x00, // USAGE (Undefined)
0xb2, 0x02, 0x01, // FEATURE (Data, Var, Abs, Buf)
0xc0 // END_COLLECTION
};
/ * Тут описано тільки один report, через що не потрібно використовувати
report-ID (він повинен бути першим байтом).
* З його допомогою передамо 3 байти даних (розмір одного REPORT_SIZE = 8
біт = 1 байт, їх кількість REPORT_COUNT = 3).
/ * Ці змінні зберігають статус поточної передачі * /
static uchar currentAddress;
static uchar bytesRemaining;
/ * UsbFunctionRead () викликається коли хост запрошувати порцію даних від
пристрою
* Для додаткової інформації див. Документацію в usbdrv.h
uchar usbFunctionRead (uchar * data, uchar len)
{
    if (len > bytesRemaining)
        len = bytesRemaining;
    uchar * buffer = (uchar *) & pdata;
    if (! currentAddress) // Жоден шматок даних ще не прочитаний.
    { // Заповнимо структуру для передачі
        if (PINB & _BV (1))
            pdata.b1 = 1;
        else
            pdata.b1 = 0;
        if (PINB & _BV (2))
            pdata.b2 = 1;
        else
            pdata.b2 = 0;
        if (PINB & _BV (3))
            pdata.b3 = 1;
        else

```

```

        pdata.b3 = 0;
    }
    uchar j;
    for (j = 0; j <len; j ++)
        data [j] = buffer [j + currentAddress];
    currentAddress + = len;
    bytesRemaining - = len;
    return len;
}
/ * UsbFunctionWrite () викликається коли хост відправляє порцію даних до
пристрою
* Для додаткової інформації див. Документацію в usbdrv.h
uchar usbFunctionWrite (uchar * data, uchar len)
{
    if (bytesRemaining == 0)
        return 1; / * Кінець передачі * /
    if (len > bytesRemaining)
        len = bytesRemaining;
    uchar * buffer = (uchar *) & pdata;
    uchar j;
    for (j = 0; j <len; j ++)
        buffer [j + currentAddress] = data [j];
    currentAddress + = len;
    bytesRemaining - = len;
    if (bytesRemaining == 0) // Всі дані отримані
    { // Виставимо значення на PORTB
        if (pdata.b1)
            PORTB | = _BV (1);
        else
            PORTB & = ~ _BV (1);
        if (pdata.b2)
            PORTB | = _BV (2);
        else
            PORTB & = ~ _BV (2);
        if (pdata.b3)
            PORTB | = _BV (3);
        else
            PORTB & = ~ _BV (3);
    }
    return bytesRemaining == 0; / * 0 означає, що є ще дані * /
}
usbMsgLen_t usbFunctionSetup (uchar data [8])

```

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

{
    usbRequest_t * rq = (void *) data;
    if ((rq-> bmRequestType & USBRQ_TYPE_MASK) == USBRQ_TYPE_CLASS) {/ *
HID пристрій * /
        if (rq-> bRequest == USBRQ_HID_GET_REPORT) {/ * wValue: ReportType
(highbyte), ReportID (lowbyte) * /
            // Тільки один різновид репорт, можемо ігнорувати report-ID
            bytesRemaining = sizeof (struct dataexchange_t);
            currentAddress = 0;
            return USB_NO_MSG; // Використовуємо usbFunctionRead () для відправки даних
хосту
        } Else if (rq-> bRequest == USBRQ_HID_SET_REPORT) {
// Тільки один різновид репорт, можемо ігнорувати report-ID
            bytesRemaining = sizeof (struct dataexchange_t);
            currentAddress = 0;
            return USB_NO_MSG; // Використовуємо usbFunctionWrite () для отримання
даних від хоста
        }
    } Else {
        / * Інші запити ми просто ігноруємо * /
    }
    return 0;
}

int main (void)
{
    DDRB = 0b00001110; // PB1, PB2, PB3 - вихід
    usbInit ();
    usbDeviceDisconnect (); // Примусово відключаємося від хоста, так
робити можна тільки при виключених переривання
    uchar i = 0;
    while (- i) {// пауза> 250 ms
        _delay_ms (1);
    }
    usbDeviceConnect (); // Підключаємося
    sei (); // Дозволяємо переривання
    for (;;) {// головний цикл програми
        usbPoll (); // Цю функцію треба регулярно викликати з головного
циклу, максимальна затримка між викликами - 50 ms
    }
    return 0;
}

```

						ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			41

Компілюємо, заливаємо прошивку в мікроконтролер, підключаємо до USB порту комп'ютера

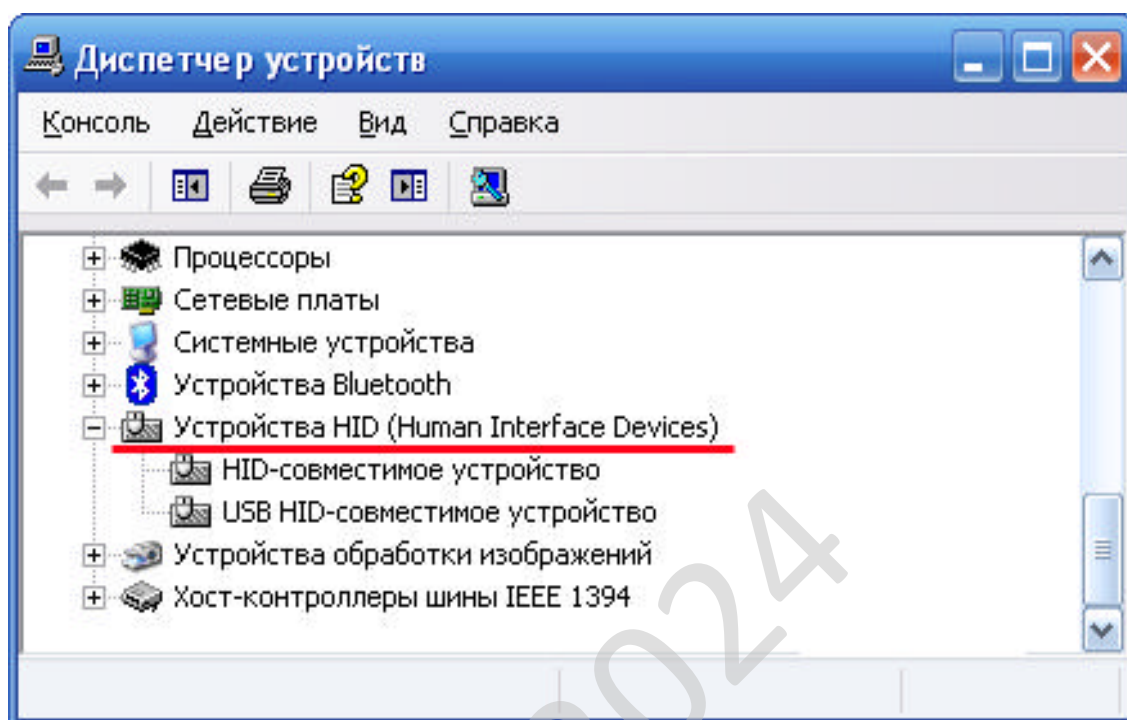


Рисунок 4.5 – Підключення до ПК

Пишемо програму для комп'ютера

Запускаємо Borland C++ Builder 6, додаємо на форму три CheckBox-а і дві звичайних кнопки:

Зберігаємо проект і копіюємо в його каталог корисну бібліотеку hidlibrary.h, вона спеціально написана для роботи з Hid пристроями з Borland C++ Builder. Зроблена як клас, також застосовуються шаблони. Бібліотека за допомогою WINAPI функції LoadLibrary підвантажує hid.dll, потім GetProcAddress – обчислює адреси функцій всередині hid.dll. Надає зручний і простий інтерфейс для роботи з HID.

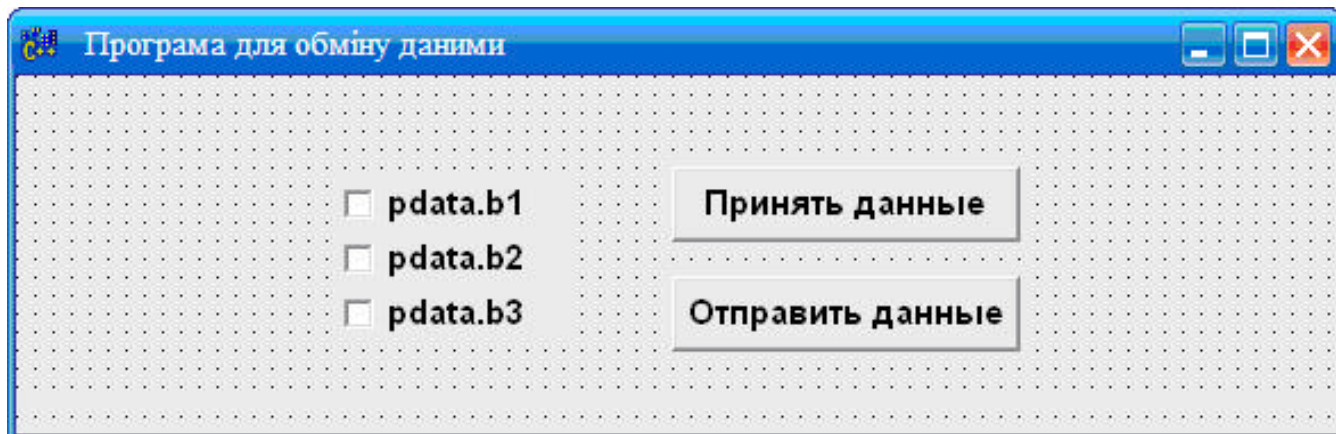


Рисунок 4.6 – Програма для обміну даними

Описана одна функція connect () для підключення нашого пристрою, і два обробника подій при натисканні на кнопки «Прийняти дані» і «Відправити дані». Все тепер можна бачити світлодіодами. Для передачі даних спеціально використовувалася структура dataexchange_t, точно така описана для МК. Щоб відкомпілювати програма потім працювала на комп'ютерах де не встановлено Borland C ++ Builder необхідно в настройках проекту відключити використання динамічних бібліотек. Для цього йдемо в Project -> Options -> вкладка Linker і Packages -> знімаємо галочки навпроти «Use dynamic RTL» і «Build with runtime packages»

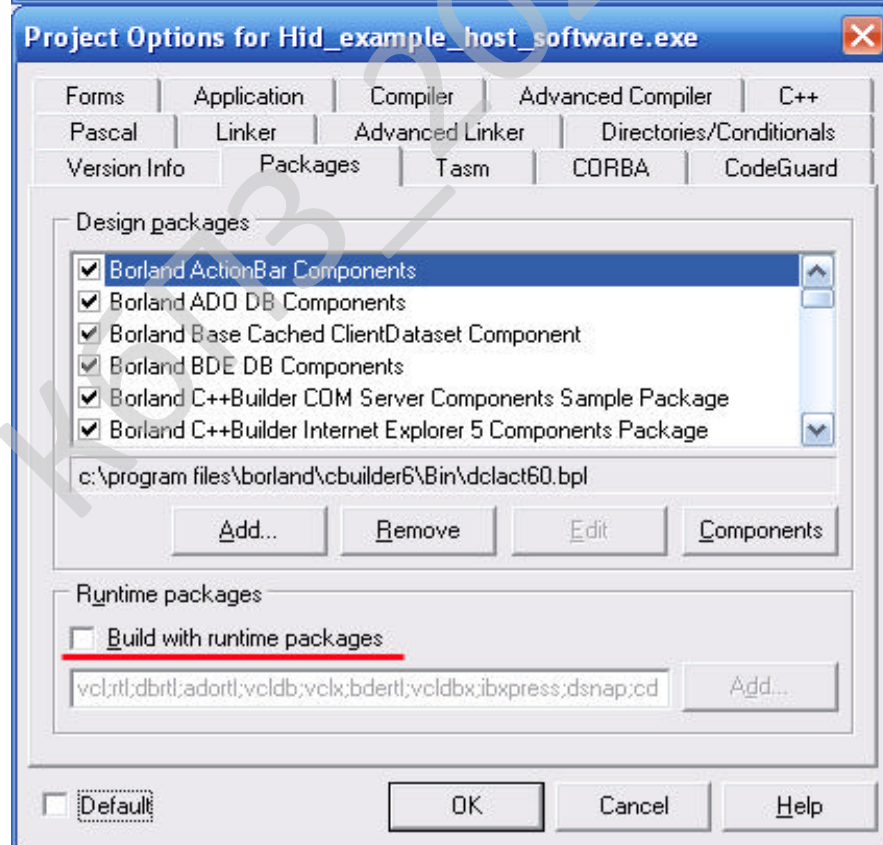
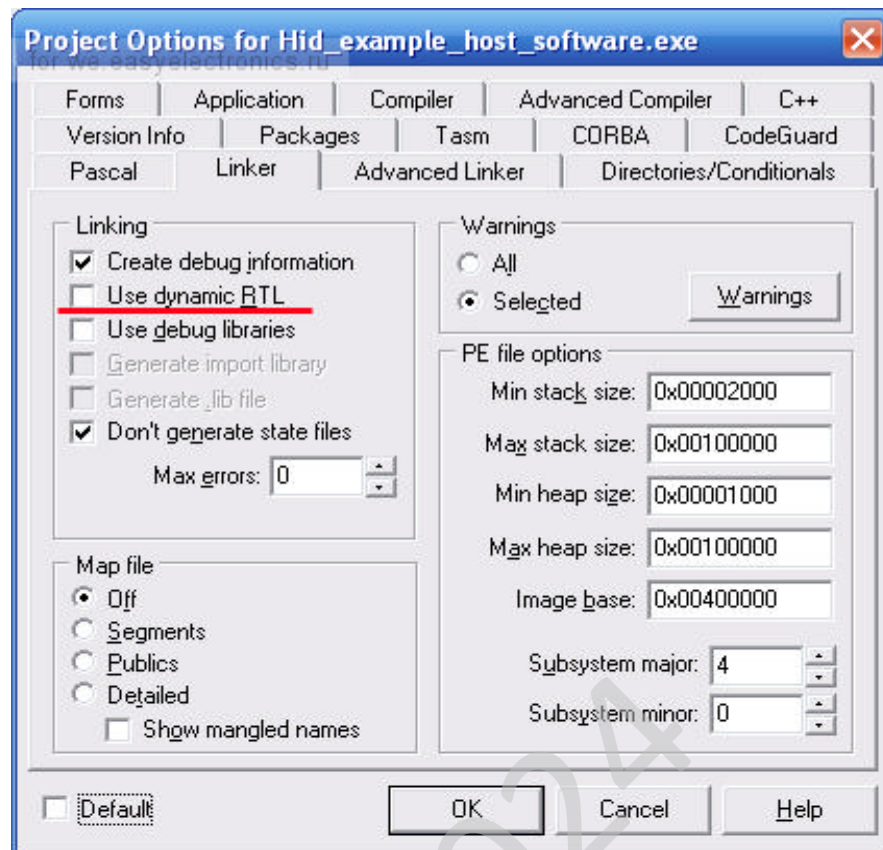


Рисунок 4.7 – Налаштування програми

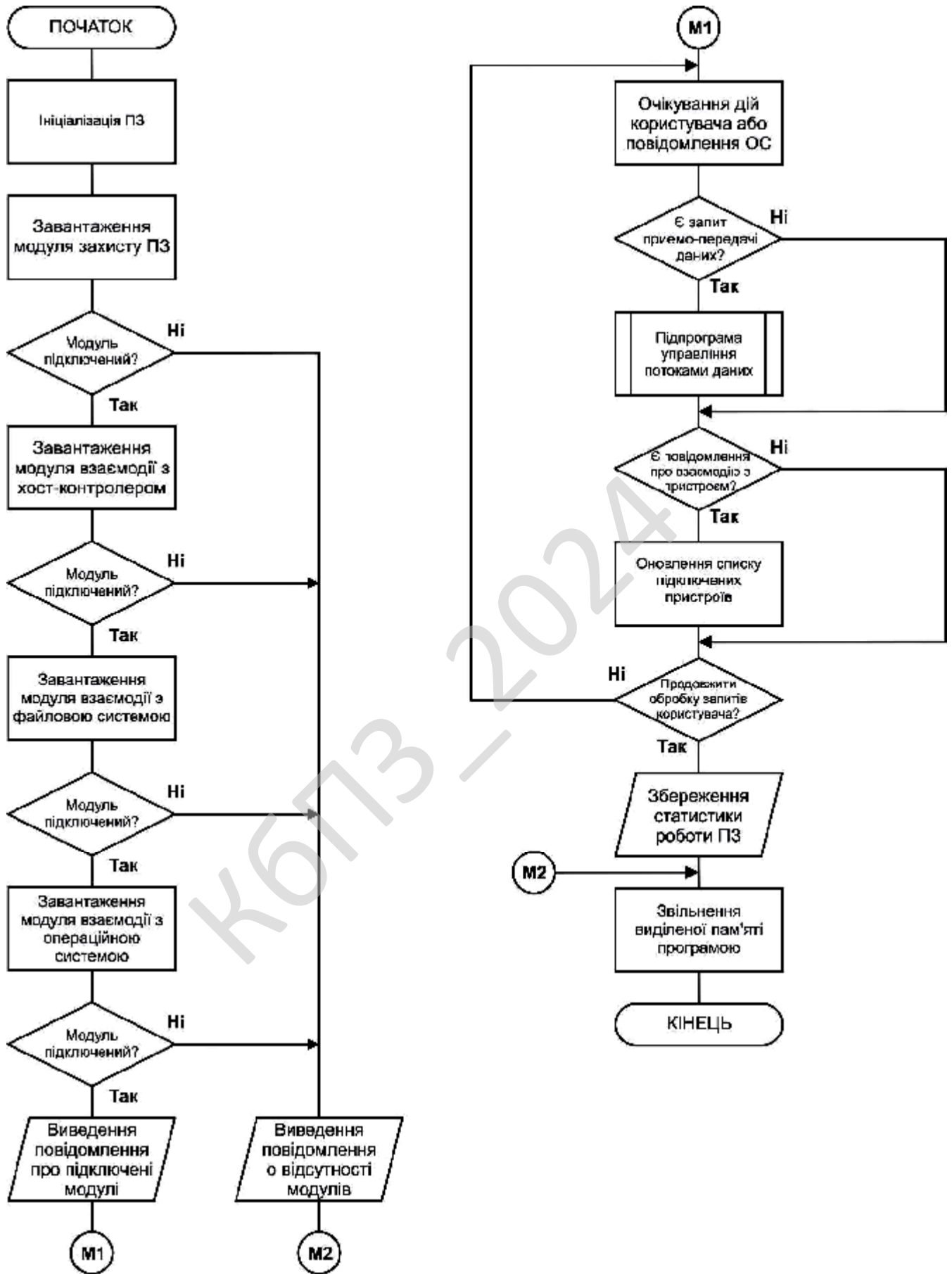


Рисунок 4.8 – Блок схема роботи програми

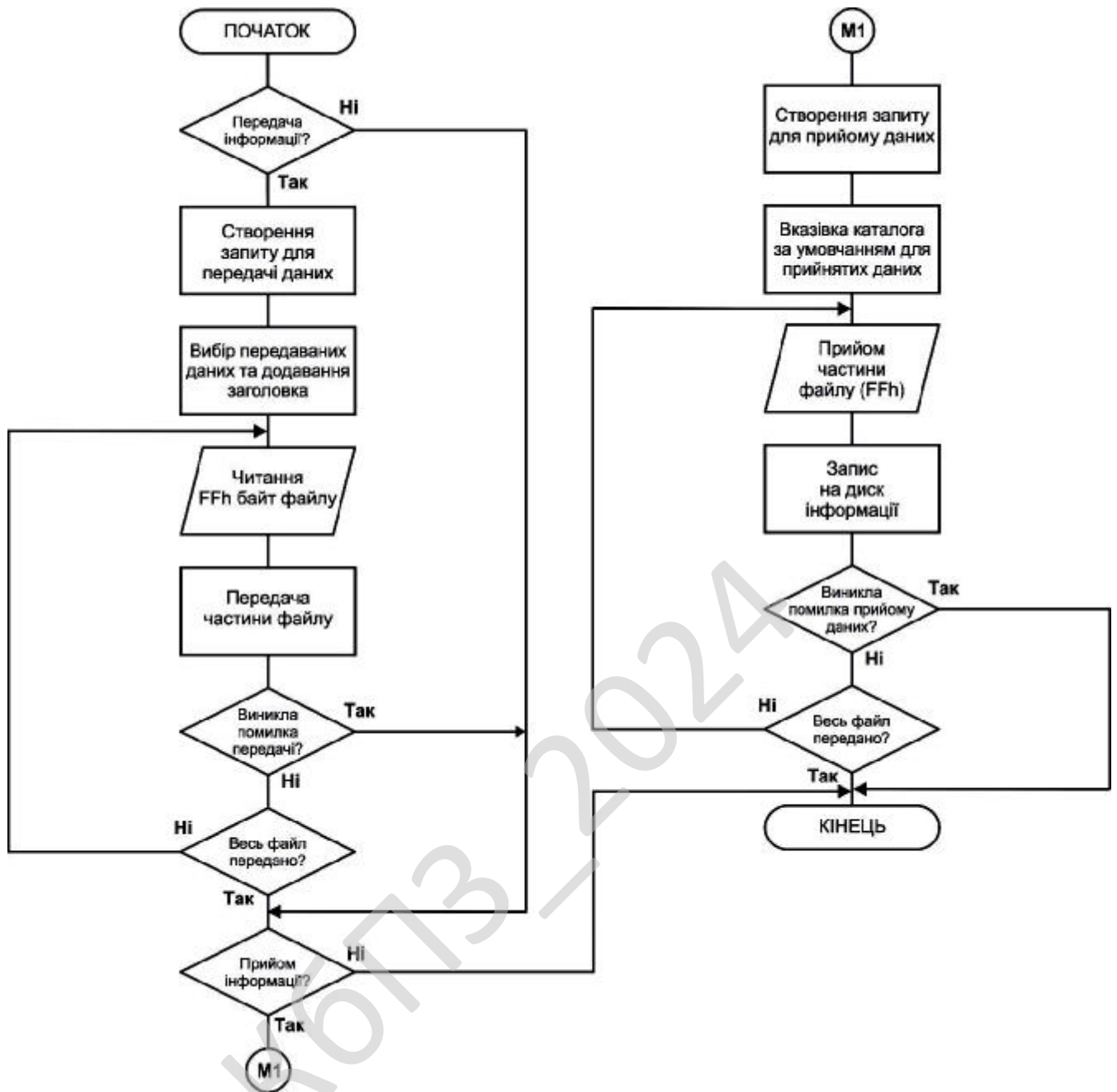


Рисунок 4.9 – Блок схема роботи підпрограми

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Serpent – симетричний блочний алгоритм шифрування, розроблений Россом Андерсоном, Елі Біхамом та Ларсом Кнудсенем. Алгоритм був одним з фіналістів 2-го етапу конкурсу AES. Як і інші алгоритми, які брали участь у

конкурсі AES, Serpent має розмір блоку 128 біт і можливі довжини ключа 128, 192 або 256 біт. Алгоритм являє собою 32-раундовий шифр на основі SP-системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, і працює з блоком з чотирьох 32-бітових слів. Serpent був розроблений так, що всі операції можуть бути виконані паралельно, використовуючи 32-а 1-бітних «потоків».

При розробці Serpent використовувався консервативніший підхід до безпеки, ніж у інших фіналістів AES, проектувальники шифру вважали, що 16 раундів достатньо, щоб протистояти відомим видам криптоаналізу, але збільшили число раундів до 32, щоб алгоритм міг краще протистояти ще не відомим методам криптоаналізу.

Шифр Serpent не запатентований і є громадським надбанням.

Алгоритм створювався під гаслом «криптографічний алгоритм 21 століття» для участі в конкурсі AES. При створенні нового алгоритму Serpent його автори дотримувалися консервативних поглядів на проектування, що підтверджується первісним рішенням про використання таблиць підстановки з відомого багато років раніше алгоритму шифрування DES, який протягом довгого часу вивчався провідними фахівцями в області криптографії та захисту інформації і чий властивості і особливості були добре відомі науковому світу. Одночасно з цим до нового алгоритму міг бути застосований вичерпний аналіз, вже розроблений для DES. Не використовувалися нові, неперевірені і невипробувані технології при створенні шифру, який у разі прийняття був би використаний для захисту величезних масивів фінансових транзакцій та урядової інформації. Основною вимогою до учасників конкурсу AES було те, що алгоритм-претендент повинен бути швидшим, ніж 3DES, і надавати як мінімум такий же рівень безпеки: він повинен мати блок даних довжиною 128 біт і ключ завдовжки 256 біт. 16-раундовий Serpent був би таким же надійним, як 3DES, але в два рази швидшим. Однак, автори вирішили, що для більшої надійності варто

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

збільшити кількість раундів до 32. Це зробило шифр таким же швидким, як DES, і набагато надійнішим, ніж 3DES.

Структура алгоритму

Алгоритм Serpent є SP-мережею, у котрій весь блок даних довжиною 128 біт на кожному раунді розбивається на 4 слова довжиною 32 біти. Всі значення, що використовуються при шифруванні є бітовими потоками. Бітові індекси змінюють значення від 0 до 31 для 32-бітових слів, від 0 до 127 – для 128-бітових блоків та від 0 до 255 для 256-бітових ключів тощо. Для внутрішніх обчислень всі біти величин представлені в прямому порядку (little-endian).

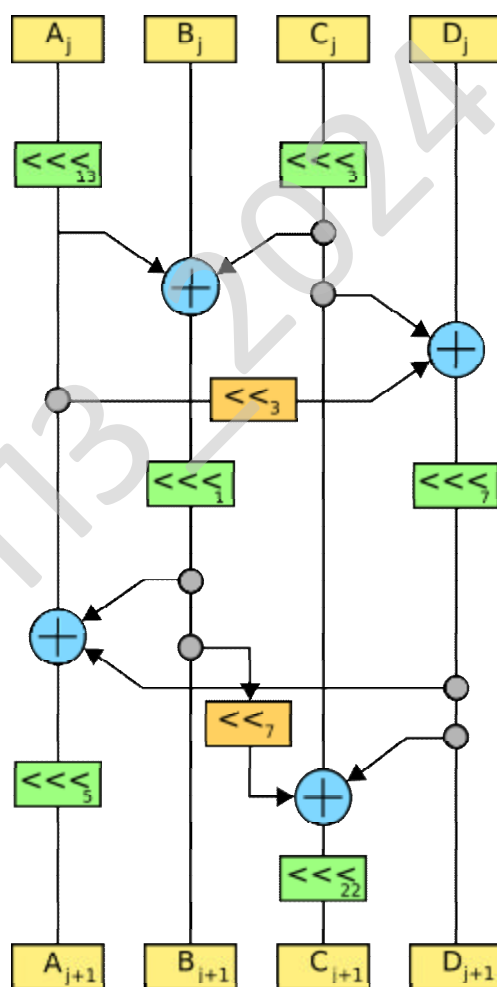


Рисунок 4.10 – Структура алгоритму Serpent

Serpent шифрує відкритий текст P довжиною 128 біт в шифротекст C довжиною таких же 128 біт за 32 раунд за допомогою 33 підключів $K\{0\}, \dots, K\{32\}$, довжиною 128 біт. Довжина використовуваного ключа може приймати різні значення, але для конкретики зафіксуємо їх довжину в 128, 192 або 256 біт. Короткі ключі довжиною менше 256 біт доповнюються до повної довжини в 256 біт.

Шифрування складається з наступних основних кроків:

- Початкова перестановка.
- 32 раунд, кожен з яких складається з операції змішування з 128-бітовим ключем (побітове логічне виключаюче «або»), таблична заміна (S-box) і лінійне перетворення. В останньому раунді лінійне перетворення замінюється додатковим накладанням ключа.

- Кінцева перестановка.

Початкова і кінцева перестановки не мають будь-якої криптографічного значущості. Вони використовуються для спрощення оптимізованої реалізації алгоритму і підвищення обчислювальної ефективності.

Розширення ключа

Як і інші алгоритми, що брали участь в конкурсі AES, Serpent має можливі довжини ключа 128, 192 або 256 біт. «Неповний» ключ довжиною менше 256 біт доповнюється за наступним правилом: додається одиничний біт справа, за ним слід стільки нульових бітів, щоб довжина ключа стала дорівнює 256 бітам.

Початкова перестановка IP

Дана перестановка IP задається таблицею, де вказується позиція, на яку перейде відповідний біт (наприклад, біт 1 перейде на 32 позицію):

S-бокси (таблиці замін)

В алгоритмі Serpent таблиці замін є 4-бітовими перестановками з властивостями стійкості до диференціального криптоаналізу, до лінійного криптоаналізу і такою властивістю, що порядок вихідних біт, як функції вхідних повинен бути максимальний, тобто бути рівним 3.

Таблиця підстановки генерується з відомих і добре вивчених таблиць для алгоритму DES в ітераційному процесі, поки не будуть отримані бажані диференціальні й лінійні властивості. Таким чином, створюється 8 таблиць підстановки.

Лінійне перетворення LT

Лінійне перетворення LT задається таблицею, де біти перераховані від 0 до 127 (наприклад, вихідний 2 біт утворений 2, 9, 15, 30, 76, 84, 126 битами, складеними за модулем 2). В кожному рядку описується 4 вихідних біти, які разом складають вхідні дані на одну таблицю заміни в наступному раунді. Варто зазначити, що даний набір являє собою таблицю $IP(LT(FP(x)))$, де LT і є те лінійне перетворення.

Таблиця зворотного лінійного перетворення, яке використовується при розшифровці ІЛТ.

Кінцева перестановка FP

Дана перестановка є зворотною до початкової, тобто $FP=IP^{-1}$ і задається наступною таблицею.

Ефективна реалізація алгоритму

Бажання авторів зробити алгоритм саме таким, яким він є стає зрозумілим при розгляді його ефективної низькорівневої реалізації.

Serpent був створений таким чином, щоб всі операції в процесі шифрування і розшифрування одного блоку могли бути виконані паралельно в 32 потоках. До того ж низькорівневий опис алгоритму набагато простіший, ніж стандартний опис. Ніяких початкових і кінцевих перестановок не потрібно.

Шифрування складається з 32 раундів. Відкритий текст є першими проміжними даними $V_0 = P$. Потім виконується 32 раунди, кожен і-й раунд складається з:

– Змішування з ключем. Проводиться побітове виключаюче «або» проміжних даних V_i з ключем довжиною 128 біт.

– Застосування таблиць підстановки. Вхідні дані довжиною 128 біт поділяються на 4 слова по 32 біта. Таблиця підстановки, реалізована послідовністю логічних операцій (як якщо це було б реалізовано апаратно), застосовується до цих 4 словам. В результаті виходить 4 вихідних слова. Таким чином, центральний процесор виконує підстановку по 32 копій таблиці одночасно.

– Лінійне перетворення. 32-бітові слова перетворюються заданим порядком.

Першою причиною вибору такого лінійного перетворення є максимізація лавинного ефекту. Такі таблиці підстановки мають властивість, що зміна кожного вхідного біта призведе до зміни 2 вихідних бітів. Таким чином, кожен вхідний біт відкритого тексту вже через 3 раунди впливає на всі вихідні біти. Аналогічно кожен біт ключа впливає на результат шифрування.

Друга причина полягає в простоті перетворення. Воно може бути реалізоване на будь-якому сучасному процесорі з мінімальними витратами.

КБПЗ-2024

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблена система має зручний та інтуїтивно зрозумілий інтерфейс. Після запуску програми на екрані з'являється вікно зображене на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна мобільного ПЗ розподілено на наступні розділи:

- Розділ функціональних кнопок;
- Розділ налаштування байтів.

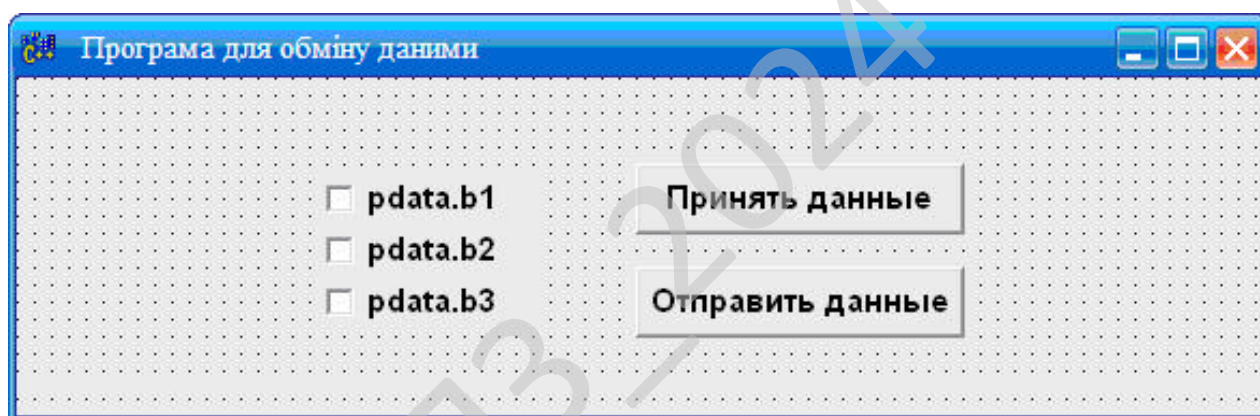


Рисунок 5.1 – Головне вікно програми

В програмі використовується дві кнопки для прийому та відправки даних. Кнопку прийняти дані легко використовується, потрібно лише вибрати з якого байта b1, b2, b2 хочемо прийняти інформацію, вибираємо ПКМ та натискаємо на кнопку .

Після цього на стенді буде загоратись діод певний діод, який був вибраний

Також аналогічно можливо відправити дані вибравши байт, а потім натискаємо на кнопку . Потім у вікні буде виведений стан операції.

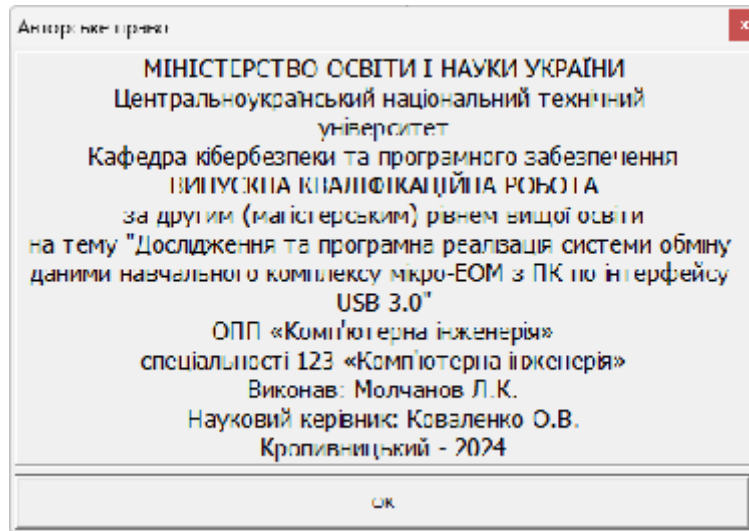


Рисунок 5.2 – Авторське право

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

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

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

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Метою розробки є дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Об'єктом дослідження є процес обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Предметом дослідження є методи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Методи дослідження базуються на методах комп'ютерної інженерії, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

– Розроблено вітчизняний продукт обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 можуть зацікавити аудиторію, схематично подану на рисунку 7.1.

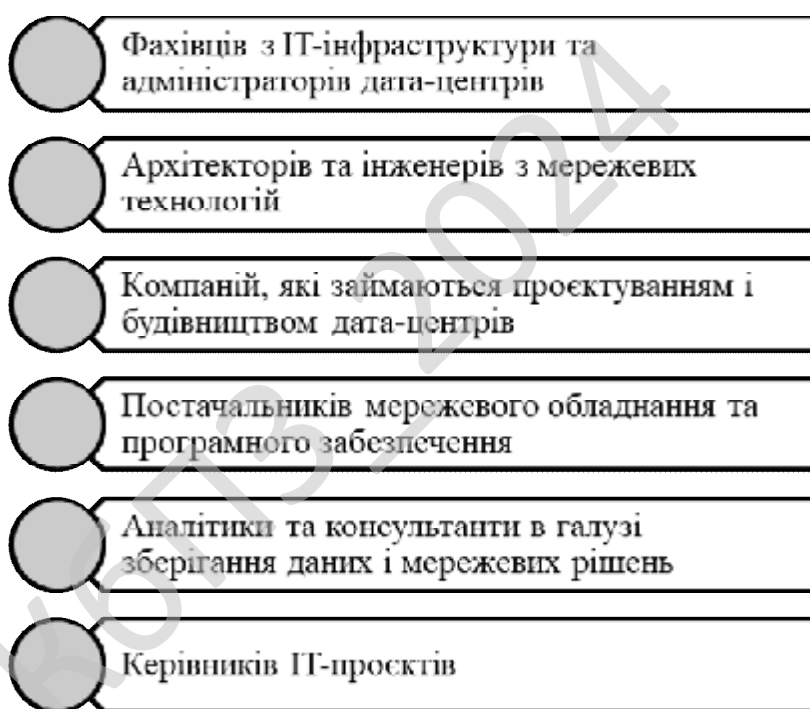


Рисунок 7.1 – Ключові сегменти цільової аудиторії

Ці зацікавлені сторони можуть отримати користь від дослідження, щоб покращити свої рішення для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, забезпечити відповідність сучасним стандартам та збільшити ефективність мережевої інфраструктури.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 можна застосувати метод експертних оцінок із застосуванням бальної шкали.

За критерії обираємо:

- надійність і безперервність роботи системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0;
- стійкість до збоїв;
- гнучкість топології (масштабованість);
- простота інтеграції з іншими системами;
- економічна ефективність;
- зручність в управлінні та адмініструванні;
- підтримка сучасних стандартів безпеки.

Експертами обираємо фахівців, які мають досвід роботи з мікро-ЕОМ та ІТ-інфраструктурою: архітекторів мереж, системних адміністраторів, аналітиків і технічних директорів.

Встановлюємо шкалу оцінювання, від 1 до 5, де:

- 1 – дуже низька привабливість,
- 5 – дуже висока привабливість.

Кожен експерт оцінює систему за всіма критеріями, присвоюючи бали відповідно до своїх знань і досвіду. Після того, як усі експерти виставили свої оцінки, обчислюємо середнє значення для кожного критерію, щоб визначити загальний рівень привабливості.

Оцінки кожного критерію аналізуємо і за їхніми результатами складаємо загальний рейтинг привабливості розробленої системи. Якщо певні критерії мають низькі оцінки, це можна вказувати на аспекти, що потребують покращення.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 7.1 – Зведені результати опитування

Критерії	Експерт 1	Експерт 2	Експерт 3	Середній бал
Надійність і безперервність	5	4	5	4.67
Стійкість до збоїв	4	5	5	4.67
Гнучкість топології	5	4	4	4.33
Простота інтеграції	4	3	4	3.67
Економічна ефективність	3	4	3	3.33
Зручність в управлінні	5	4	4	4.33
Підтримка стандартів безпеки	5	5	5	5.00
Загальний середній бал: 4.33				

На основі отриманих даних робимо висновок про високий рівень привабливості системи для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0. Водночас варто звернути увагу на економічну ефективність, яка потребує вдосконалення. Такий підхід дозволяє якісно оцінити потенційну привабливість системи на етапі планування та доопрацювання.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 варто вибрати

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

метод, який враховує специфічність проекту, витрати на ресурси та необхідні технічні вимоги. Варто обирати між методом оцінки на основі функціональних точок, методом оцінки вартості на основі витрат та методом аналогів. Для програмної реалізації системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, де важлива точна оцінка функціоналу і витрат на ресурси, метод оцінки на основі функціональних точок (FPA) або метод витрат (Cost-Based Estimation) можуть бути оптимальними. Але все ж метод аналогів рекомендуємо як над швидкий. Адже є аналогічні проекти із подібними функціональними та технічними вимогами. Це швидкий метод, який саме і варто застосувати.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічна ефективність від впровадження системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 даних може проявлятися через скорочення витрат, оптимізацію роботи системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 та покращення загальної продуктивності.

Зменшення витрат на проектування та розгортання системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 першочергово проявляється через можливість швидко створювати і тестувати різні варіанти топології без додаткових витрат на інженерів та проектувальників та зниження потреби в повторних проектних і монтажних роботах завдяки точності та оптимізації початкового проекту. Розрахунково витрати на проектування і тестування мережевої топології можна зменшити на 20–30%.

Оптимізація використання обладнання шляхом аналізу та визначення найбільш ефективних маршрутів і конфігурації, що допомагає зменшити кількість обладнання або замінити дороге обладнання на більш ефективні

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

альтернативи; зменшення капітальних витрат на обладнання через зниження кількості непотрібних мережевих пристроїв. Розрахунково на 10%.

Скорочення операційних витрат завдяки зниженню кількості простоїв. Оптимальна топологія зменшує ризики простоїв та проблем із продуктивністю, що мінімізує витрати на підтримку і забезпечення безперебійності роботи. Зменшення простоїв збільшує ефективний час роботи сервісів, що знижує ризик втрати доходів через незаплановані зупинки. В середньому на 25%.

Зменшення витрат на енергоспоживання. Оптимізовані мережеві рішення дозволяють скоротити кількість енергоємного обладнання та знизити загальне енергоспоживання. Системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, спроектовані з урахуванням енергоефективності, можуть знизити витрати на електроенергію для дата-центру. Не менше ніж на 10%.

Зменшення витрат на навчання персоналу (до 50 %). Автоматизація і спрощення роботи з мережевою інфраструктурою дозволяє скоротити витрати на навчання та адаптацію персоналу. Замість того щоб проводити спеціалізовані навчання з кожної нової мережевої конфігурації, персонал швидше адаптується до нової системи. Покращення якості обслуговування клієнтів. Стабільність та гнучкість мережевої інфраструктури підвищують якість обслуговування кінцевих користувачів. Задоволеність клієнтів допомагає утримувати поточних клієнтів і залучати нових, що позитивно впливає на дохідність компанії. Покращення якості обслуговування на 10% може зменшити відтік клієнтів на 5%.

Економічна ефективність досягається щорічно завдяки зменшенню операційних витрат, оптимізації витрат на проектування, зниженню енергоспоживання та підвищенню продуктивності мережевої інфраструктури.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Таблиця 7.2 – Економічна ефективність від реалізації

ПРОЯВ ЕФЕКТИВНОСТІ	КІЛЬКІСНЕ ЗНАЧЕННЯ
Зменшення витрат на проектування та розгортання системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0	Скорочення витрат на 20-30%
Оптимізація використання обладнання шляхом аналізу та визначення найбільш ефективних маршрутів і конфігурації	До 10%
Скорочення операційних витрат	В середньому на 25%
Зменшення витрат на енергоспоживання	Не менше ніж на 10%.
Зменшення витрат на навчання персоналу	до 50 %
Покращення якості обслуговування клієнтів	Покращення якості обслуговування на 10% може зменшити відтік клієнтів на 5%

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Покроковий алгоритм для просування проекту програмної реалізації системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 представлено графічно на рисунку 7.2. Цей алгоритм орієнтований на залучення цільової аудиторії та побудову довіри до продукту.

Завдяки комплексному підходу до просування проекту можна ефективно розширити ринок, залучити цільову аудиторію та створити довіру до продукту.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проекту програмної системи для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0, можна застосувати наступні стратегії:

- напряду пропонувати продукт великим дата-центрам і компаніям з високими вимогами до інфраструктури (телекомунікаційні оператори, великі підприємства тощо);

- встановити партнерські стосунки з компаніями, які надають консалтингові або інтеграційні послуги для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0;

- використовувати точкову рекламу на професійних платформах, таких як linkedin, де зосереджені фахівці, пов'язані з керуванням дата-центрами;

- співпраця з відомими експертами в галузі системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 та публікації на спеціалізованих ресурсах можуть забезпечити довіру до продукту і підвищити його видимість;

- публікація матеріалів, статей і кейсів, які вирішують реальні проблеми проектування та оптимізації мереж у дата-центрах, що допоможе покращити позиції у пошукових системах і привернути увагу цільових клієнтів;

- надавати можливість інтеграції з уже існуючими рішеннями для управління мережевою інфраструктурою.

- запровадження гнучкої передплатної моделі (місячної, річної тощо) зробить систему доступною для ширшого кола клієнтів, які можуть випробувати її можливості без значних початкових інвестицій;

- розробити програми для швидкого освоєння системи, що підвищить рівень задоволення користувачів і сприятиме довготривалим відносинам з клієнтами;

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

1. Аналіз ринку та конкурентів	Оцінка ринкової ніші: дослідити, як інші рішення задовольняють потреби дата-центрів у проєктуванні мереж.
	Визначення унікальних переваг (USP): сформулювати ключові особливості, які відрізняють продукт від конкурентів (наприклад, зменшення витрат, простота інтеграції, гнучкість).
	Аналіз цільової аудиторії: визначити основних споживачів – адміністратори дата-центрів, мережеві архітектори, інженери.
2. Позиціонування та брендинг	Розробка повідомлення бренду: створити чітке та зрозуміле повідомлення, що підкреслює ефективність системи в автоматизації та оптимізації мережевої топології.
	Розробка логотипу та візуальних матеріалів: створити візуальні матеріали, що будуть асоціюватися з продуктом.
	Розробка сайту і лендінгу продукту: підготувати сайт із описом продукту, його перевагами, кейсами і формою для заявок.
3. Маркетингові дослідження та аналіз попиту	Опитування та інтерв'ю з потенційними клієнтами: щоб краще зрозуміти очікування та основні потреби клієнтів.
	Створення демо-версії продукту: розробити демонстраційну версію або віртуальний тур для потенційних клієнтів, щоб вони могли ознайомитися з можливостями системи.
4. Контент-маркетинг	Написання статей та white papers: публікувати матеріали про переваги оптимізації мережевої інфраструктури в ЦОДах, використовуючи приклади економічної ефективності та надійності.
	Блог на веб-сайті: ділитися результатами впровадження, кейсами та інсайтами у сфері автоматизації проєктування мереж.
	Вебінари та навчальні матеріали: організувати вебінари для показу можливостей продукту, створення записів із прикладами використання.
5. Соціальні мережі та професійних спільнот	LinkedIn та Twitter: публікувати анонси, новини та статті про продукт, а також брати участь в обговореннях на професійних платформах.
	GitHub та форуми мережевих технологій: якщо є відкриті компоненти продукту або бібліотеки, варто публікувати їх на GitHub і залучати спільноту до обговорень та тестування.
6. PR та участь у виставках	Прес-релізи: анонсувати вихід продукту на спеціалізованих ресурсах та у виданнях, орієнтованих на IT-інфраструктуру та технології дата-центрів.
	Участь у виставках та конференціях: представляти продукт на конференціях з тематики дата-центрів, таких як Data Center World, де є потенційні клієнти та партнери.
	Публікації у технічних журналах та на порталах: готувати матеріали для ресурсів, що спеціалізуються на технологіях дата-центрів та мережевих рішеннях.
7. Сформування партнерств	Співпраця з постачальниками мережевого обладнання та програмного забезпечення: запропонувати інтеграцію та оптимізацію їхнього обладнання з вашою системою.
	Альянси з консалтинговими компаніями: співпрацювати з компаніями, які надають послуги із проєктування дата-центрів і можуть рекомендувати вашу систему своїм клієнтам.
8. Програма тестування для ключових клієнтів	Плотний запуск у партнерських організаціях: запропонувати безкоштовне тестування або зв'язки для перших клієнтів, щоб зібрати зворотний зв'язок і показати переваги рішення.
	Відгуки та кейси: зібрати відгуки і створити реальні кейси, що демонструють позитивні зміни після впровадження продукту.
9. Оцінка та оптимізація стратегії просування	Аналіз даних просування: регулярно аналізувати відгуки, відвідуваність вебсайту, конверсії та залучення аудиторії в соціальних мережах.
	Коригування маркетингової стратегії: на основі зібраної інформації та відгуків клієнтів змінювати акценти просування, наприклад, більше наголошувати на тих перевагах продукту, які найбільше цінуються ринком.
10. Післяпродажна підтримка та постійне вдосконалення	Служба підтримки клієнтів: створити команду технічної підтримки для швидкого реагування на запити користувачів.
	Регулярні оновлення та нові функції: забезпечувати стабільні оновлення, пропонувати нові функціональні можливості та вдосконалення на основі відгуків клієнтів.

Рисунок 7.2 – Алгоритм просування проєкту

– забезпечити високоякісну технічну підтримку та допомогу з налаштуванням системи, що зменшить ризики та підвищить рівень довіри до продукту;

– використовувати систему CRM для відстеження ефективності різних каналів та прийняття рішень щодо оптимізації роботи з партнерами та клієнтами;

– завдяки CRM проводити сегментацію за потребами, наприклад, між тими, хто потребує простих рішень для малих ЦОДів, і тими, хто розгортає складні, масштабовані системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0;

– випуск оновлень продукту на основі зворотного зв'язку допомагає втримувати клієнтів і створює позитивний імідж продукту як того, що активно розвивається.

Ці заходи допоможуть ефективно оптимізувати канали збуту, підвищити конкурентоспроможність продукту і розширити його ринкове охоплення серед центрів обробки даних та системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи проєктування та побудови топології мереж для центрів обробки даних та системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 наведені на рисунку 7.3.

Виконання кожного з цих факторів допоможе забезпечити конкурентоспроможність і високу привабливість системи для ринку, а також стійке зростання клієнтської бази.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

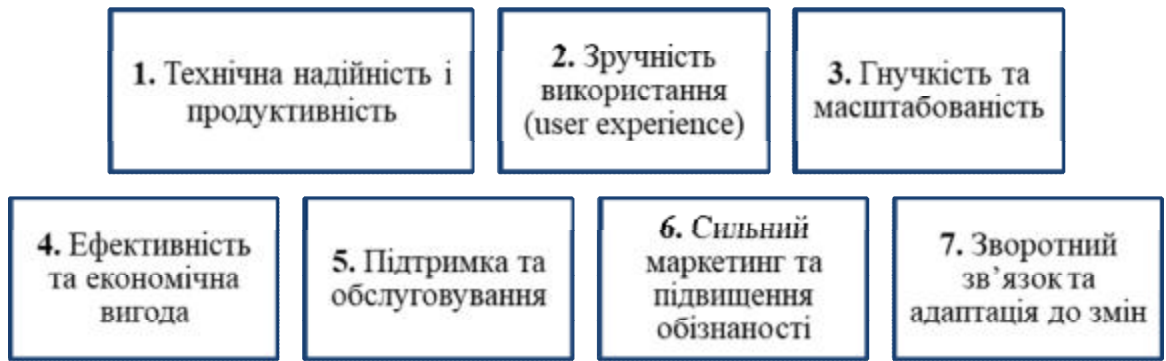


Рисунок 7.3 – Ключові фактори успіху проєкту

КБПЗ_2024

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Програмісти у процесі роботи отримують негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки (м'язи рук та суглоби пальців) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій спеціалісти відносять високочастотні електромагнітні коливання роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці [5].

8.2 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 пожежною та вибухонебезпекою приміщення відноситься до класу В. В таблиці 8.1 наведена загальна характеристика приміщення щодо пожежної та вибухонебезпеки та важкістю робіт.

Таблиця 8.1. – Загальна характеристика приміщення щодо вибухопожежонебезпеки 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 І-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 [6].

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 [6] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		67

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК

Розглянемо приміщення в якому працює користувач ПК з даним програмним продуктом.

Приміщення має одностороннє природне освітлення і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита темним ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря в нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS монітором, а також меблі.

Приміщення має довжину 4м, ширину 3,5 м, висоту стелі 2,7 м. Кількість робочих місць – одне. Площа – 14 м², об'єм – 37,8 м³. Виходячи з цього, отримано дані, наведені в таблиці 8.2.

Таблиця 8.2 – Фактичні та нормативні значення параметрів приміщення

Параметр	Норма *	Реальні параметри
Площа, S	не менше 6 м ²	14 м ²
Об'єм, V	не менше 20 м ³	37,8 м ³

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

За даними, які наведено у табл. 8.2, можна зробити висновок, що отримані показники, площа та об'єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Щодо мікроклімату, то згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [7] роботу з ПК можна віднести до категорії легка 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання взимку. Оператором виділяється до 120ккал теплової енергії за

годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 8.3.

Таблиця 8.3 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 ⁰ С	24 ⁰ С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24 ⁰ С	23 ⁰ С
	Вологість	40 – 60%	55%
	Швидкість повітря	< 0,1м/с	

*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

По отриманим замірам параметрів мікроклімату можна зробити висновок, що усі показники задовольняють вимогам зазначеним для робіт категорії легка 1а і є задовільними для здоров'я людини.

Щодо освітлення, то згідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [7] ця робота відноситься до V_a розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає $S' = 1,8 \cdot 1,5 = 2,7 \text{ м}^2$ і є боковим освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком лампи 900 лм. Згідно замірів рівень освітлення в даному приміщенні і на робочому місці складає в межах 350-500 лк, що відповідає нормованому значенню

Джерелом шуму в приміщенні є комп'ютер. Вентилятори (кулери) системного блоку, процесора, відеокарти і блоку живлення є сучасними і мають

низький рівень шуму. Згідно з технічною документацією шум, зумовлений кулером в блоці живлення складає 25 дБ, кулером процесора – 30 дБ, загальний - 34 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера і незначний рівень фонового шуму від іншого устаткування, можна стверджувати, що сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБА, що відповідає рівню шуму для приміщень з комп'ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

У приміщенні відсутні джерела інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК вироблений на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновією лампою, що не має сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

8.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропоную використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості, проведених відділом охорони праці.

Як міри по зниженню шуму можна запропонувати:

- облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);
- екранування робочого місця (постановкою перегородок, діафрагм);
- установка в комп'ютерних приміщеннях устаткування, що робить мінімальний шум;
- раціональне планування приміщення.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

8.5 Розрахункова частина

При роботі з комп'ютерною та офісною технікою головними причинами ураження електричним струмом є двофазне дотикання; однофазне дотикання; дотик до корпусу обладнання, яке не проводить струм, але опинилося під напругою; перебування в зоні дії атмосферної або статичної електрики; вхід у зону дії електромагнітного поля.

Правильна організація експлуатації й обслуговування комп'ютерної та офісної техніки, контрольно-вимірювальної апаратури оговорена "Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів" [8]. Розрахуємо конструкцію захисного заземлення (рис. 8.1) за методикою [9, 10].

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

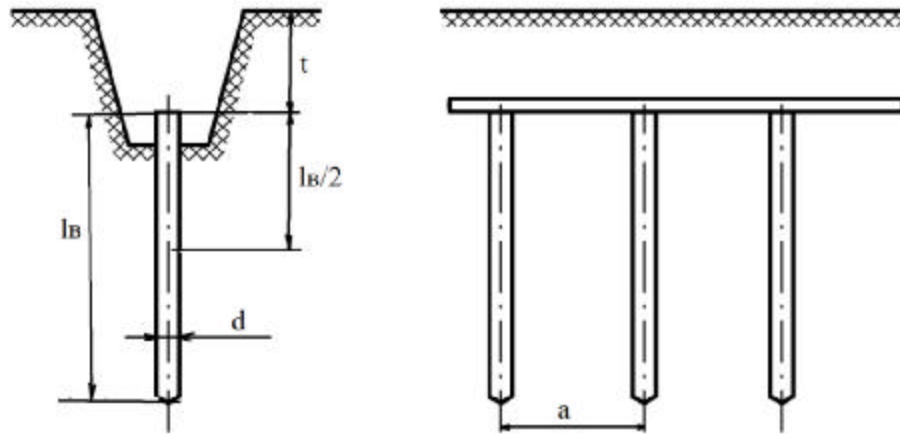


Рисунок 8.1 – Схема електричного захисного заземлення

Початкові дані для розрахунку:

- напруга живлення 380 В;
- режим нейтралі трансформатора – ізольована; – природне заземлення присутнє, $R_e = 20 \text{ Ом}$;
- ґрунт – глина; питомий опір – $\rho_r = 60 \text{ Ом}\cdot\text{м}$;
- вологість глини під час вимірювання її опору – нормальна; – кліматична зона – третя.

Відповідно рекомендацій ПУЕ умови виконання завдання визначають значення

$$R_{nye} = 4 \text{ Ом.}$$

Природні заземлювачі наявні, тому допустимий опір R_d штучних заземлювачів

$$R_d \leq \frac{20 \cdot 4}{20 - 4} \leq 5 \text{ Ом,}$$

Для виконання заземлення обираємо вертикальні електроди із кутової сталі шириною профілю $b_k = 50 \text{ мм}$, довжиною $l_v = 3 \text{ м}$; для таких вертикальних електродів коефіцієнт сезонності $\psi = 1,3$;

- відстань між електродами $a = 3 \text{ м}$, тобто $a / l_v = 1$;
- горизонтальний з'єднувальний електрод – стальна смуга шириною $b_g =$

$$n_3 = \frac{21,424}{5 \cdot 0,635} \approx 7.$$

Таким чином отримали, що кількість електродів не змінюється (різниця між останніми числами заземлювачів менше одиниці). Тобто число вертикальних заземлювачів складає 7.

Визначаємо довжину горизонтального електроду. Для заземлювачів, розташованих у ряд

$$L_g = 1,05 \cdot 1 \cdot (7 - 1) = 18,9 \text{ м.}$$

Знаходимо опір розтікання струму горизонтального електроду, визначивши спочатку величину ρ_2 , $\psi = 2,5$:

$$\rho_2 = 60 \cdot 2,5 = 150 \text{ Ом}\cdot\text{м},$$

$$R_g = \frac{150}{2\pi \cdot 18,9} \ln \frac{2 \cdot 18,9 \cdot 18,9}{0,045 \cdot 0,8} = 12,5 \text{ Ом}$$

Визначаємо загальний результуючий опір системи заземлення, спочатку визначивши коефіцієнт використання горизонтального електроду за допомогою табличними даними. Для цього, виконуючи лінійну інтерполяцію, напишемо рівняння:

$$\eta_g = -0,00667 \cdot n_3 + 0,7933 = 0,747;$$

$$R_{gr} = \frac{21,424 \cdot 12,5}{21,424 \cdot 0,747 + 7 \cdot 12,5 \cdot 0,636} = 3,742 \text{ Ом.}$$

Порівнюємо значення R_{gr} з R_d . Співвідношення $R_{gr} < R_d$ виконується, тобто вибрана конструкція заземлення задовольняє вимогам ПУЕ.

Таким чином, приймаємо наступну конструкцію заземлення: вертикальні електроди розміщені в ряд в кількості 7 штук, довжина з'єднувального горизонтального електроду складає 18,9 м.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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 обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.
- Досліджена система обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.
- На основі отриманих результатів досліджень створена програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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 Borland 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 Serpent.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Молчанов Л.К. Дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0 // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.

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

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

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

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

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

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

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

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

10. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

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

13. 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.

14. 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.

15. 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.

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

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

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

24. 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.

25. 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*

					BKPM-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

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

					БКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

34. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів ІЕС60880 та ІЕС62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

36. 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.

37. 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.

38. 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.

39. 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.*

40. 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.*

41. 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*

42. 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.*

43. 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.*

44. 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.

45. 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.*

46. 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.*

					ВКРМ-123.24.0023.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

47. 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.

48. 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).

49. 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

50. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

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

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-123.24.0023.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Молчанов Л.К.				Дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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. № 19-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0.

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-123.24.0023.00.00.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 обміну даними навчального комплексу мікро-ЕОМ з ПК по інтерфейсу USB-3.0;
- 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-123.24.0023.00.00.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, 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 Borland C++.

					ВКРМ-123.24.0023.00.00.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 маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК.

					ВКРМ-123.24.0023.00.00.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 – 83 аркуші.

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 Маркетингове та економічне обґрунтування ІТ-проєкту.

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 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2024 р.

					ВКРМ-123.24.0023.00.00.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

_____ Коваленко О.В.

*Дослідження та програмна реалізація
системи обміну даними навчального комплексу мікро-ЕОМ з ПК по
інтерфейсу USB-3.0*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 13

Літера: РП

Кропивницький – 2024 року

Бібліотека для роботи з Hid на Borland C ++ Builder

```

const char idstring [] = "vid_16c0 & pid_05df";

#include <windows.h>
#include <setupapi.h>
#include <string>
#include <vector>
#include <map>

typedef void (WINAPI * t_HidD_GetHidGuid) (OUT LPGUID);
typedef BOOLEAN (WINAPI * t_HidD_GetManufacturerString) (IN HANDLE, OUT
PVOID, IN ULONG);
typedef BOOLEAN (WINAPI * t_HidD_GetProductString) (IN HANDLE, OUT PVOID,
IN ULONG);
typedef BOOLEAN (WINAPI * t_HidD_GetFeature) (IN HANDLE, OUT PVOID, IN
ULONG);
typedef BOOLEAN (WINAPI * t_HidD_SetFeature) (IN HANDLE, IN PVOID, IN
ULONG);

using namespace std;

template <typename T> class HIDLibrary
{
protected:
static const int datasize = sizeof (T);
T data;
char pad [16];
GUID m_hidguid;
HINSTANCE hDLL;
t_HidD_GetHidGuid HidD_GetHidGuid;
t_HidD_GetManufacturerString HidD_GetManufacturerString;
t_HidD_GetProductString HidD_GetProductString;
t_HidD_GetFeature HidD_GetFeature;
t_HidD_SetFeature HidD_SetFeature;
vector <string> m_HIDPaths;
map <string, string> m_HIDDeviceIdents;
string GetDevicePath (HDEVINFO hInfoSet, PSP_DEVICE_INTERFACE_DATA
oInterface);
string m_ConnectedDevice;
public:
HIDLibrary ();
bool IsAvailableLib ();
int EnumerateHIDDevices ();
bool Connect (unsigned int device = 0);
string GetConnectedDevicePath () {return m_ConnectedDevice;};
string GetConnectedDeviceName () {return m_HIDDeviceIdents
[m_ConnectedDevice];};
string GetDeviceCount () {return m_HIDPaths.size (); };
string GetDeviceName (int i) {
if (i <m_HIDPaths.size ())
return m_HIDDeviceIdents [m_HIDPaths [i]];
else
return string.empty;
}
int SendData (T * data);
int ReceiveData (T * data);
int IsConnectedDevice ();
};

template <typename T> HIDLibrary <T> :: HIDLibrary ()
{
hDLL = LoadLibrary ("HID.DLL");
if (hDLL! = NULL)
{
HidD_GetHidGuid = (t_HidD_GetHidGuid) GetProcAddress (hDLL,
"HidD_GetHidGuid");
}
}

```

```

        HidD_GetManufacturerString = (t_HidD_GetManufacturerString)
GetProcAddress (hDLL, "HidD_GetManufacturerString");
        HidD_GetProductString = (t_HidD_GetProductString) GetProcAddress (hDLL,
"HidD_GetProductString");
        HidD_GetFeature = (t_HidD_GetFeature) GetProcAddress (hDLL,
"HidD_GetFeature");
        HidD_SetFeature = (t_HidD_SetFeature) GetProcAddress (hDLL,
"HidD_SetFeature");
        if (HidD_GetHidGuid)
        {
            HidD_GetHidGuid (& m_hidguid);
        }
    }

    template <typename T> bool HIDLibrary <T> :: IsAvailableLib ()
    {
        return! (hDLL == NULL || HidD_GetHidGuid == NULL ||
HidD_GetManufacturerString == NULL ||
        HidD_GetProductString == NULL || HidD_GetFeature == NULL ||
HidD_SetFeature == NULL);
    }

    template <typename T> string HIDLibrary <T> :: GetDevicePath (HDEVINFO
hInfoSet, PSP_DEVICE_INTERFACE_DATA oInterface)
    {
        char vpath [1000];
        vpath [0] = 0;
        DWORD nRequiredSize = 0;
        // Get the device interface details
        if (! SetupDiGetDeviceInterfaceDetail (hInfoSet, oInterface, 0, 0, &
nRequiredSize, 0))
        {
            PSP_DEVICE_INTERFACE_DETAIL_DATA oDetail =
(PSP_DEVICE_INTERFACE_DETAIL_DATA) malloc (nRequiredSize);
            oDetail-> cbSize = sizeof (SP_DEVICE_INTERFACE_DETAIL_DATA);
            if (SetupDiGetDeviceInterfaceDetail (hInfoSet, oInterface, oDetail,
nRequiredSize, & nRequiredSize, 0))
            {
                string s (oDetail-> DevicePath);
                free (oDetail);
                return s;
            }
        }
        return "";
    }

    template <typename T> int HIDLibrary <T> :: EnumerateHIDDevices ()
    {
        wchar_t buffer [1000];
        char s1 [1000];
        m_HIDPaths.clear ();
        int nIndex = 0;
        HDEVINFO hInfoSet = SetupDiGetClassDevs (& m_hidguid, 0,0,
DIGCF_DEVICEINTERFACE | DIGCF_PRESENT);
        SP_DEVICE_INTERFACE_DATA spd;
        spd.cbSize = sizeof (SP_DEVICE_INTERFACE_DATA);
        while (SetupDiEnumDeviceInterfaces (hInfoSet, 0, & m_hidguid,
nIndex, & spd))
        {
            string path = GetDevicePath (hInfoSet, & spd);
            if (path.find (idstring) != string ::npos)
            {
                HANDLE h = CreateFile (path.c_str (), GENERIC_READ, 0,0, OPEN_EXISTING,
FILE_FLAG_OVERLAPPED, 0);
                if (h != INVALID_HANDLE_VALUE)
                {
                    HidD_GetManufacturerString (h, buffer, 500);
                    wcstombs (s1, buffer, 1000);
                }
            }
        }
    }

```

```

    if (strlen (s1) > 2)
    {
        strcat (s1, "");
        HidD_GetProductString (h, buffer, 500);
        wcstombs (s1 + strlen (s1), buffer, 1000);
    }
    CloseHandle (h);
}
else
    strcpy (s1, "Unknown device");
string name (s1);
m_HIDDeviceIdents [path] = s1;
m_HIDPaths.push_back (path);
}
nIndex ++;
}
SetupDiDestroyDeviceInfoList (hInfoSet);
return nIndex;
}

template <typename T> bool HIDLibrary <T> :: Connect (unsigned int
device)
{
    if (m_HIDPaths.size () < device + 1)
    {
        EnumerateHIDDevices ();
        if (m_HIDPaths.size () < device + 1) return false;
    }
    m_ConnectedDevice = m_HIDPaths [device];
    return true;
}

template <typename T> int HIDLibrary <T> :: SendData (T * data)
{
    char vpath [datasize + 16];
    vpath [0] = 0;
    int len = datasize + 1;
    HANDLE h = CreateFile (m_ConnectedDevice.c_str (), GENERIC_READ, 0, 0,
OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);
    if (h != INVALID_HANDLE_VALUE)
    {
        memcpy (vpath + 1, data, datasize);
        int err = HidD_SetFeature (h, vpath, len);
        CloseHandle (h);
        return err;
    }
    else
        return 0;
}

template <typename T> int HIDLibrary <T> :: ReceiveData (T * data)
{
    char vpath [datasize + 16];
    memset (vpath, 0, sizeof (vpath));
    int len = datasize + 1;
    HANDLE h = CreateFile (m_ConnectedDevice.c_str (), GENERIC_READ, 0, 0,
OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);
    if (h != INVALID_HANDLE_VALUE)
    {
        int err = HidD_GetFeature (h, vpath, len);
        memcpy (data, vpath + 1, datasize);
        CloseHandle (h);
        return err;
    }
    else
        return 0;
}

template <typename T> int HIDLibrary <T> :: IsConnectedDevice ()

```

```

{
    HANDLE h = CreateFile (m_ConnectedDevice.c_str (), GENERIC_READ, 0,0,
OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);
    if (h! = INVALID_HANDLE_VALUE)
    {
        CloseHandle (h);
        return 1;
    }
    else
        return 0;
}
Драйвер на usb 3.0
/ * Ім'я: usbconfig.h
* /

#ifndef __usbconfig_h_included__
#define __usbconfig_h_included__

/ * Основне опис:
* Цей файл є прикладом конфігурації (з вбудованою документацією) для
* Драйвера USB. Він кофігуріруєт V-USB для сигналу USB D +, приєднаного
до
* Порту D біт 2 (який також є входом апаратного переривання 0 [INT0]
* На багатьох чіпах) і сигналу USB D- до порту D біт 4. Ви можете
під'єднати
* Ці сигнали на будь-який інший порт, однак D + повинен бути також
підключений на
* Вхід апаратного переривання INT0, яке має найвищий пріоритет,
* См. Секцію в кінці файлу).
* /

/ * ----- Апаратні настройки -----
--- * /

#define USB_CFG_IOPORTNAME D
/ * Вказано порт, до якого підключена шина USB. Якщо Ви сконфігуріруєте
"В",
* Будуть використовуватися регістри PORTB, PINB і DDRB.
* /
#define USB_CFG_DMINUS_BIT 4
/ * Це номер біта в USB_CFG_IOPORT, куди підключений сигнал D-, може
* Використовуватися будь біт в порте.
* /
#define USB_CFG_DPLUS_BIT 2
/ * Це номер біта в USB_CFG_IOPORTNAME, куди підключений сигнал D +, може
* Використовуватися будь біт в порте. Будь ласка, візьміть до уваги, що
D +
* Повинен бути також приєднаний до ніжки переривання INT0! [Ви можете
також
* Використовувати інші переривання, див. Секцію "Optional MCU
Description" далі,
* Або Ви можете під'єднати D- до переривання, як це необхідно якщо Ви
* Використовуєте опцію USB_COUNT_SOF. Якщо Ви використовуєте D- для
переривання,
* Воно буде спрацьовувати також від маркерів Start-Of-Frame кожду
* Мілісекунди.]
* /
#define USB_CFG_CLOCK_KHZ (F_CPU / 1000)
/ * Тактова частота чіпа AVR в кілогерц. Допустимі величини 12000, 15000,
* 16000, 16500 і 20000. Версія коду на 16.5 МГц не вимагає кварцу, вона
* Припускає відхід частоти +/- 1% від номінальної. Всі інші частоти
вимагають
* Точності 2000 ppm, і в цьому випадку обов'язково потрібен кварц!
* Константа F_CPU повинна бути визначена в проекті, не потрібно ніде
писати
* Вручну.
* /
#define USB_CFG_CHECK_CRC 0

```

```

/ * Встановить 1 якщо хочете щоб драйвер V-USB перевіряв цілісність
входять
* Пакетів даних (CRC суми). Перевірка CRC вимагає досить багато місця
для
* Коду і на даний момент підтримується тільки для частоти 18 MHz. Так що
* Обов'язково використовуйте частоту 18 MHz для включення цієї опції.
* /

/ * #define USB_CFG_PULLUP_IOPORTNAME D * /
/ * Якщо Ви під'єднали 1.5k pullup резистор від D- до ніжки порту замість
V +,
* Ви можете приєднувати і від'єднувати пристрій програмним шляхом
виклику
* Макрос usbDeviceConnect () і usbDeviceDisconnect () (див. Usbdrv.h).
* Ця константи задають порт, до якого підключений pullup резистор.
* /
/ * #define USB_CFG_PULLUP_BIT 4 * /
/ * This constant defines the bit number in USB_CFG_PULLUP_IOPORT
(defined
* Above) where the 1.5k pullup resistor is connected. See description
* Above for details.
* Ця константа визначає номер біта порту в USB_CFG_PULLUP_IOPORTNAME
(заданий
* Више) куди підключений 1.5k pullup резистор. Див. Опис, вказане
раніше.
* /

/ * ----- Функціональний діапазон -----
-- - * /

#define USB_CFG_HAVE_INTRIN_ENDPOINT 1
/ * Задайте тут 1, якщо Ви хочете скомпілювати версію з двома кінцевими
* Точками (endpoints): default control endpoint 0 і interrupt-in
endpoint
* (Будь-який інший endpoint номер).
* /
#define USB_CFG_HAVE_INTRIN_ENDPOINT3 0
/ * Задайте тут 1, якщо Ви хочете скомпілювати версію з трьома кінцевими
* Точками (endpoints): default control endpoint 0, interrupt-in endpoint
3
* (Або номер, конфігурований далі) і catch-all default interrupt-in
* Endpoint як зазначено раніше.
* Ви повинні також задати USB_CFG_HAVE_INTRIN_ENDPOINT в 1 для цієї
опції.
* /
#define USB_CFG_EP3_NUMBER 3
/ * Якщо так звана кінцева точка (endpoint) 3 використовується, можна
тепер
* Її конфігурувати на будь-який інший номер кінцевої точки (за винятком
0).
* Якщо не задано, то за замовчуванням 3.
* /
/ * #define USB_INITIAL_DATATOKEN USBPID_DATA1 * /
/ * Цей define визначає умову старту (startup condition) для перемикання
* Даних (data toggling) в момент кінцевих точок interrupt / bulk 1 і 3.
* За замовчуванням USBPID_DATA1. Оскільки токен перемикається ПЕРЕД
відправкою
* Будь-яких даних, перший пакет відправляється с протилежної величиною
* Від цієї конфігурації!
* /
#define USB_CFG_IMPLEMENT_HALT 0
/ * Задайте тут 1, якщо Ви хочете також реалізувати можливість
ENDPOINT_HALT
* Для endpoint 1 (interrupt endpoint). Незважаючи на те, що Вам ця
можливість
* Може не знадобитися, вона потрібна за стандартом. Ми ввели цю опцію в
* Конфігурацію тому що ця функція значно роздуває код.
* /
#define USB_CFG_SUPPRESS_INTR_CODE 0

```

7

```

/ * Задайте тут 1, якщо Ви хочете визначити interrupt-in endpoints, але
не
* Хочете відправляти будь-які дані з їх допомогою. Якщо ця опція
включена, то
* Функції usbSetInterrupt () і usbSetInterrupt3 () будуть пропущені.
Корисно
* При необхідності interrupt-in endpoints для соответствия інтерфейсу
* (Наприклад HID), але без відправки даних з їх допомогою. Пристойно
економить
* Флеш пам'ять для коду та оперативну пам'ять буфера передачі даних.
* /
#define USB_CFG_INTR_POLL_INTERVAL 100
/ * Якщо Ви скопіювали версію з endpoint 1 (interrupt-in), то тут
задається
* Інтервал опитування (poll interval). Величина вказана в мілісекундах і
не повинна
* Бути менше 10 ms для низькошвидкісних пристроїв.
* /
#define USB_CFG_IS_SELF_POWERED 0
/ * Задайте тут 1 якщо пристрій має власний джерело живлення.
* Установка в 0 відповідає пристрою, живити від шини USB.
* /
#define USB_CFG_MAX_BUS_POWER 100
/ * Встановіть цю змінну на max величину споживання потужності Вашого
* Пристрою від шини USB. Величина вказана в міліамперах.
* /
#define USB_CFG_IMPLEMENT_FN_WRITE 1
/ * Задайте тут 1, якщо Ви хочете, щоб функція usbFunctionWrite ()
* Викликала для передач control-out. Задайте 0 якщо Вам це не
потрібно,
* Що дозволить заощадити кілька байт.
* /
#define USB_CFG_IMPLEMENT_FN_READ 1
/ * Задайте 1 якщо Вам потрібно відправляти керуючі запити відповіді, які
* Генеруються "на льоту" при виклику usbFunctionRead (). Якщо Вам
потрібно тільки
* Відправити дані з статичного буфера, задайте 0 і повертайте дані з
* UsbFunctionSetup (). Це заощадить деяку кількість байт.
* /
#define USB_CFG_IMPLEMENT_FN_WRITEOUT 0
/ * Задайте 1 якщо Ви хочете використовувати interrupt-out (або bulk out)
* Endpoints. Ви повинні реалізувати функцію usbFunctionWriteOut (), яка
* Приймає всі interrupt / bulk дані, відправлені в будь-яку endpoint,
* Відмінну від 0. Номер endpoint можна знайти в 'usbRxToken'.
* /
#define USB_CFG_HAVE_FLOWCONTROL 0
/ * Задайте 1, якщо Ви хочете реалізувати управління потоком
(flowcontrol)
* Через дані USB. Дивимося опис макросу usbDisableAllRequests () і
* UsbEnableAllRequests () в usbdrv.h.
* /
#define USB_CFG_DRIVER_FLASH_PAGE 0
/ * Якщо пристрій має більше ніж 64 kBytes флеш пам'яті, визначте це
* В 64 k сторінку де знаходяться константи (дескриптори) драйвера.
Іншими
* Словами: встановити в 1 для завантажувачів (boot loaders) на
ATMega128.
* /
#define USB_CFG_LONG_TRANSFERS 0
/ * Задайте 1, якщо Ви хочете відправляти / приймати блоки оазмером більш
254 байт
* В одиночній control-in або control-out передачі. Зверніть увагу, що ця
* Можливість збільшує розмір драйвера.
* /
/ * #define USB_RX_USER_HOOK (data, len) if (usbRxToken == (uchar)
USBPID_SETUP) blinkLED (); * /
/ * Цей макро є гаком (hook), якщо Ви хочете зробити щось
* Нетрадиційне. Якщо макро заданий, він вставляється в початок обробки

```

```

* Прийнятого повідомлення. Якщо Ви "з'їли" (обробили) повідомлення і не
хочете
* Включати подальшу обробку за замовчуванням, зробіть звичайний
повернення після
* Ваших дій. Одне з можливих застосувань (крім налагодження) - миготіння
* Статусним світлодіодом при кожному пакеті.
* /
/ * #define USB_RESET_HOOK (resetStarts) if (! ResetStarts) {hadUsbReset
();} * /
/ * Цей макро є хуком для обробки події USB RESET. Він має один
* Параметр, що дозволяє відрізнити старт стану RESET від закінчення
* Стану RESET.
* /
/ * #define USB_SET_ADDRESS_HOOK () hadAddressAssigned (); * /
/ * Цей макро (якщо заданий) виконується коли прийнято запит USB
SET_ADDRESS.
* /
#define USB_COUNT_SOF 0
/ * Задайте тут 1, якщо Вам потрібна глобальна змінна "usbSofCount",
* У якій підраховуються пакети SOF. Ця можливість вимагає підключення
* Апаратного переривання до D- замість D +.
* /
/ * #ifdef __ASSEMBLER__
* Macro myAssemblerMacro
* In YL, TCNT0
* Sts timer0Snapshot, YL
* Endm
* #endif
* #define USB_SOF_HOOK myAssemblerMacro
* This macro (if defined) is executed in the assembler module when a
* Start Of Frame condition is detected. It is recommended to define it
to
* The name of an assembler macro which is defined here as well so that
more
* Than one assembler instruction can be used. The macro may use the
register
* YL and modify SREG. If it lasts longer than a couple of cycles, USB
messages
* Immediately after an SOF pulse may be lost and must be retried by the
host.
every
* What can you do with this hook? Since the SOF signal occurs exactly
in
* 1 ms (unless the host is in sleep mode), you can use it to tune OSCCAL
the
* Designs running on the internal RC oscillator.
* Please note that Start Of Frame detection works only if D- is wired to
* Interrupt, not D +. THIS IS DIFFERENT THAN MOST EXAMPLES!
* /
#define USB_CFG_CHECK_DATA_TOGGLING 0
/ * Встановіть в 1 якщо хочете фільтрувати повторювані (duplicate) пакети
помилки
* Даних відправлені хостом. Повторення виникають лише в результаті
самим у
* Язку, коли хост не отримав АСК. Пам'ятайте, фільтрувати доведеться
* UsbFunctionWriteOut () і usbFunctionWrite (). Використовуйте глобальну
* UsbCurrentDataToken і статичну змінну для кожного
* Control- і out-endpoint для перевірки на повторювані пакети.
* /
#define USB_CFG_HAVE_MEASURE_FRAME_LENGTH 0
/ * Задайте тут 1, якщо Вам потрібна функція usbMeasureFrameLength () -
тоді вона
* Скопільється. Ця функція може використовуватися для підстроювання
генератора
* RC мікроконтролера AVR.
* /
#define USB_USE_FAST_CRC 0
/ * Асемблерної частина має дві реалізації для алгоритму CRC. Одна
швидше,

```

```

* Друга менша за обсягом. Використовуйте меншу лише в передачах
повідомлень
* Не критичних до таймингам. Швидка реалізація вимагає 31 цикл на байт,
а
* Менша вимагає 61 - 69 циклів. Швидка реалізація може бути на 32 байта
* Коду більшою при передачах даних і виводить AVR на стан близький до
* Межі можливостей.
* /

/ * ----- Параметри, що описують USB-пристрій -----
-- * /

#define USB_CFG_VENDOR_ID 0xc0, 0x16 / * = 0x16c0 = 5824 = voti.nl * /
/ * USB вендор ID для пристрою, молодший байт йде першим. Якщо Ви
* Зареєстрували свій власний Vendor ID, вкажіть його тут. В
* Іншому випадку використовуйте одну з вільно надаються пар
* VID / PID obdev. Прочитайте правила USB-IDs-for-free.txt!
* /
#define USB_CFG_DEVICE_ID 0xdf, 0x05 / * obdev's shared PID for HID's * /
/ * Це ID продукту, молодший байт йде першим. Він інтерпретується в
контексті
* Vendor ID. Якщо Ви зареєстрували свій власний VID на usb.org, або
* Якщо користуєтеся ліцензією на PID від кого-небудь ще, вкажіть його
тут.
* Інакше використовуйте надану obdev вільно поширювану
* Пару VID / PID. Прочитайте правила USB-IDs-for-free.txt!
* /
#define USB_CFG_DEVICE_VERSION 0x00, 0x01
/ * Номер версії пристрою: спочатку молодший номер, потім старший номер.
* /
#define USB_CFG_VENDOR_NAME 'w', 'e', '.', 'e', 'a', 's', 'y', 'e', 'l',
'e', 'c', 't', 'r', 'o', 'n', 'i', 'c', 's', '.', 'r', 'u'
#define USB_CFG_VENDOR_NAME_LEN 21
/ * Тут вказують ім'я вендора (vendor name), що повертається пристроєм.
* Ім'я повинно надаватися як список символів в одиночних
* Лапках, а USB_CFG_VENDOR_NAME_LEN задає їх кількість. Символи
* Інтерпретуються як Unicode (UTF-16). Якщо Вам не потрібно ім'я
вендора,
* Закоментуйте цей макрос. ЗАВЖДИ вказуйте ім'я вендора, що містить Ваше
* Доменне ім'я Internet, якщо Ви використовуєте вільно поширювану пару
* Obdev VID / PID. За деталями звертайтеся до файлу USB-IDs-for-
free.txt.
* /
#define USB_CFG_DEVICE_NAME 'H', 'i', 'd', ' ', 'e', 'x', 'a', 'm', 'p',
'l', 'e'
#define USB_CFG_DEVICE_NAME_LEN 11
/ * Тут вказується ім'я пристрою (device name) таким же способом, як і в
* Попередньому параметрі вказується ім'я вендора. Якщо Вам не потрібно
ім'я
* Пристрою, закоментуйте цей макрос. Див. Файл USB-IDs-for-free.txt
* Перед призначенням імені, якщо Ви використовуєте вільно поширювану
* Пару obdev VID / PID.
* /
/ * # Define USB_CFG_SERIAL_NUMBER 'N', 'o', 'n', 'e' * /
/ * # Define USB_CFG_SERIAL_NUMBER_LEN 0 * /
/ * Серійний номер (serial number). Якщо серійний номер Вам не потрібен,
* Закоментуйте макрос. Може виявитися корисним надати serial number
* З інших міркувань, що відрізняється від часу компіляції. Див. Секцію,
* Описує властивості дескриптора далі, де вказано як зробити точне
* Управління дескрипторами USB (USB descriptors), такими як string
* Descriptor для серійного номера.
* /
#define USB_CFG_DEVICE_CLASS 0
#define USB_CFG_DEVICE_SUBCLASS 0
/ * Див. Специфікацію USB, якщо Ви хочете пристосувати пристрій до
* Існуючого класу пристроїв (device class). Клас 0xff e
* "Специфічним для вендора" ("vendor specific").
* /
#define USB_CFG_INTERFACE_CLASS 3

```

```

#define USB_CFG_INTERFACE_SUBCLASS 0
#define USB_CFG_INTERFACE_PROTOCOL 0
/ * Див. Специфікацію USB, якщо Ви хочете, щоб пристрій відповідало
вимогам
* Існуючого класу або протоколу. Наступні класи повинні бути встановлені
* На рівні інтерфейсу:
* HID class 3, Нетреба підкласів та протоколів (може виявитися
корисним!)
* CDC class 2, використовуйте підклас 2 і протокол 1 для ACM
* /
#define USB_CFG_HID_REPORT_DESCRIPTOR_LENGTH 22
/ * Задайте тут довжину HID report descriptor, якщо Ви реалізували HID
* Пристрій. Інакше не задавайте цей параметр або встановіть його в 0.
* Якщо використовуєте цей параметр, то Ви повинні додати символний
масив
* PROGMEM, названий "usbHidReportDescriptor", в код, який містить
* Report descriptor. Не забудьте тримати масив і цей параметр в
* Відповідно!
* /

/ * #define USB_PUBLIC static * /
/ * Використовуйте цей define, якщо Ви використовуєте директиву #include
usbdrv.c
* Замість лінкування. Ця техніка економить кілька байт flash-пам'яті.
* /

/ * ----- Тонке управління через дескриптори USB -----
--- * /
/ * Якщо Ви не хочете використовувати дескриптори за замовчуванням
драйвера USB,
* Ви можете надати Ваші власні. Це може бути зроблено
* Як (1) статичні дані фіксованої довжини в пам'яті flash,
* (2) статичні дані фіксованої довжини в RAM, або (3) динамічні під
* Час виконання у функції usbFunctionDescriptor (). Дивимось usbdrv.h
для
* Більшої інформації по таким дескрипторів. Підтримка дескрипторів
* Конфігурируется через властивості дескриптора. Якщо не задані
властивості або
* Якщо вони рівні 0, використовується дескриптор за замовчуванням.
* Можливі властивості:
* + USB_PROP_IS_DYNAMIC: дані для дескриптора повинні бути захоплені
(fetch)
* Під час виконання через usbFunctionDescriptor (). Якщо
використовується
* Механізм usbMsgPtr, дані за замовчуванням у FLASH.
* Додайте властивість USB_PROP_IS_RAM якщо хочете RAM покажчики.
* + USB_PROP_IS_RAM: дані, повернуті usbFunctionDescriptor () або
* Знайдені в статичної пам'яті RAM, не в пам'яті flash.
* + USB_PROP_LENGTH (len): якщо дані в статичної пам'яті (RAM або
flash),
* Драйвер повинен знати довжину дескриптора. Дескриптор сам по собі
знайдений за адресою,
* Відомому як ідентифікатор (див. Далі).
* Списку статичних імен дескрипторів (повинен бути задекларований
PROGMEM в flash):
* Char usbDescriptorDevice [];
* Char usbDescriptorConfiguration [];
* Char usbDescriptorHidReport [];
* Char usbDescriptorString0 [];
* Int usbDescriptorStringVendor [];
* Int usbDescriptorStringDevice [];
* Int usbDescriptorStringSerialNumber [];
* Інші дескриптори не можуть бути надані статично, вони повинні
* Надаватися динамічно під час виконання.
*
* Властивості дескрипторів об'єднуються за АБО небудь складаються разом,
наприклад:
* #define USB_CFG_DESCR_PROPS_DEVICE (USB_PROP_IS_RAM | USB_PROP_LENGTH
(18))

```

```

*
* Наступні дескриптори задані:
* USB_CFG_DESCR_PROPS_DEVICE
* USB_CFG_DESCR_PROPS_CONFIGURATION
* USB_CFG_DESCR_PROPS_STRINGS
* USB_CFG_DESCR_PROPS_STRING_0
* USB_CFG_DESCR_PROPS_STRING_VENDOR
* USB_CFG_DESCR_PROPS_STRING_PRODUCT
* USB_CFG_DESCR_PROPS_STRING_SERIAL_NUMBER
* USB_CFG_DESCR_PROPS_HID
* USB_CFG_DESCR_PROPS_HID_REPORT
* USB_CFG_DESCR_PROPS_UNKNOWN (for all descriptors not handled by the
driver)
*
* Пам'ятайте що string дескриптори не просто рядки, вони є
* Unicode strings prefixed with a 2 byte header. Приклад:
* Int serialNumberDescriptor [] = {
* USB_STRING_DESCRIPTOR_HEADER (6),
* 'S', 'e', 'r', 'i', 'a', 'l'
* };
* /

#define USB_CFG_DESCR_PROPS_DEVICE 0
#define USB_CFG_DESCR_PROPS_CONFIGURATION 0
#define USB_CFG_DESCR_PROPS_STRINGS 0
#define USB_CFG_DESCR_PROPS_STRING_0 0
#define USB_CFG_DESCR_PROPS_STRING_VENDOR 0
#define USB_CFG_DESCR_PROPS_STRING_PRODUCT 0
#define USB_CFG_DESCR_PROPS_STRING_SERIAL_NUMBER 0
#define USB_CFG_DESCR_PROPS_HID 0
#define USB_CFG_DESCR_PROPS_HID_REPORT 0
#define USB_CFG_DESCR_PROPS_UNKNOWN 0

/ * ----- Додаткові налаштування MCU -----
-- * /
/ * Наступні конфігурації працюють за умовчанням в usbdrv.h. Вам звичайно
не потрібно
* Встановлювати їх. Тільки якщо Вам потрібно запустити драйвер на
пристрої,
* Яке поки не обробляється компілятором, який неповністю
* Підтримується (як, наприклад, IAR C) або якщо використовуєте
переривання,
* Відмінне від INT0, Ви можете задати дещо з цього.
* /
/ * #define USB_INTR_CFG MCUCR * /
/ * #define USB_INTR_CFG_SET ((1 << ISC00) | (1 << ISC01)) * /
/ * #define USB_INTR_CFG_CLR 0 * /
/ * #define USB_INTR_ENABLE GIMSK * /
/ * #define USB_INTR_ENABLE_BIT INTO * /
/ * #define USB_INTR_PENDING GIFR * /
/ * #define USB_INTR_PENDING_BIT INTF0 * /
/ * #define USB_INTR_VECTOR INTO_vect * /

#endif / * __usbconfig_h_included__ * /

#include <avr / io.h>
#include <avr / interrupt.h>
#include <util / delay.h>
#include <avr / pgmspace.h> / * потрібно для usbdrv.h * /
#include "usbdrv.h"
struct dataexchange_t // Опис структури для передачі даних
{
    uchar b1; // Я вирішив для прикладу написати структуру на 3 байта.
    uchar b2; // На кожен байт підчепити ногу з PORTB. Звичайно це
    uchar b3; // Не раціонально (всього то 3 біти потрібно).
}; // Але з метою демонстрації в самий раз.
// Для наочності прикрутити по світодиоду і споглядати :)
struct dataexchange_t pdata = {0, 0, 0};

```

```

PROGMEM char usbHidReportDescriptor [22] = { // USB report descriptor //
Дескриптор описує структуру пакета даних для обміну
    0x06, 0x00, 0xff, // USAGE_PAGE (Generic Desktop)
    0x09, 0x01, // USAGE (Vendor Usage 1)
    0xa1, 0x01, // COLLECTION (Application)
    0x15, 0x00, // LOGICAL_MINIMUM (0) // min. значення для даних
    0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255) // max. значення для
даних, 255 тут не випадково, а щоб укластися в 1 байт
    0x75, 0x08, // REPORT_SIZE (8) // інформація передається порціями,
це розмір одного "репорт" 8 біт
    0x95, sizeof (struct dataexchange_t), // REPORT_COUNT // кількість
порцій (у нашому прикладі = 3, описана вище структура передається за три
репорт)
    0x09, 0x00, // USAGE (Undefined)
    0xb2, 0x02, 0x01, // FEATURE (Data, Var, Abs, Buf)
    0xc0 // END_COLLECTION
};
/ * Тут ми описали тільки один report, через що не потрібно
використовувати report-ID (він повинен бути першим байтом).
* З його допомогою передамо 3 байти даних (розмір одного REPORT_SIZE =
8 біт = 1 байт, їх кількість REPORT_COUNT = 3).
/ * Ці змінні зберігають статус поточної передачі * /
static uchar currentAddress;
static uchar bytesRemaining;
/ * UsbFunctionRead () викликається коли хост запрошувати порцію даних
від пристрою
* Для додаткової інформації див. Документацію в usbdrv.h
uchar usbFunctionRead (uchar * data, uchar len)
{
    if (len > bytesRemaining)
        len = bytesRemaining;
    uchar * buffer = (uchar *) & pdata;
    if (! currentAddress) // Жоден шматок даних ще не прочитаний.
    { // Заповнимо структуру для передачі
        if (PINB & _BV (1))
            pdata.b1 = 1;
        else
            pdata.b1 = 0;
        if (PINB & _BV (2))
            pdata.b2 = 1;
        else
            pdata.b2 = 0;
        if (PINB & _BV (3))
            pdata.b3 = 1;
        else
            pdata.b3 = 0;
    }
    uchar j;
    for (j = 0; j < len; j++)
        data [j] = buffer [j + currentAddress];
    currentAddress + = len;
    bytesRemaining - = len;
    return len;
}
/ * UsbFunctionWrite () викликається коли хост відправляє порцію даних
до пристрою
* Для додаткової інформації див. Документацію в usbdrv.h
uchar usbFunctionWrite (uchar * data, uchar len)
{
    if (bytesRemaining == 0)
        return 1; / * Кінець передачі * /
    if (len > bytesRemaining)
        len = bytesRemaining;
    uchar * buffer = (uchar *) & pdata;
    uchar j;
    for (j = 0; j < len; j++)
        buffer [j + currentAddress] = data [j];
    currentAddress + = len;
    bytesRemaining - = len;
}

```

```

if (bytesRemaining == 0) // Всі дані отримані
{// Виставимо значення на PORTB
    if (pdata.b1)
        PORTB |= _BV (1);
    else
        PORTB &= ~ _BV (1)
    if (pdata.b2)
        PORTB |= _BV (2);
    else
        PORTB &= ~ _BV (2);
    if (pdata.b3)
        PORTB |= _BV (3);
    else
        PORTB &= ~ _BV (3);
}
return bytesRemaining == 0; /* 0 означає, що є ще дані */
}
usbMsgLen_t usbFunctionSetup (uchar data [8])
{
    usbRequest_t * rq = (void *) data;
    if ((rq-> bmRequestType & USBRQ_TYPE_MASK) == USBRQ_TYPE_CLASS) {/
* HID пристрій */
        if (rq-> bRequest == USBRQ_HID_GET_REPORT) {/ * wValue:
ReportType (highbyte), ReportID (lowbyte) */
            // У нас тільки один різновид репорт, можемо ігнорувати
report-ID
                bytesRemaining = sizeof (struct dataexchange_t);
                currentAddress = 0;
                return USB_NO_MSG; // Використовуємо usbFunctionRead () для
відправки даних хосту
            } Else if (rq-> bRequest == USBRQ_HID_SET_REPORT) {
                // У нас тільки один різновид репорт, можемо ігнорувати
report-ID
                bytesRemaining = sizeof (struct dataexchange_t);
                currentAddress = 0;
                return USB_NO_MSG; // Використовуємо usbFunctionWrite ()
для отримання даних від хоста
            }
        } Else {
            /* Інші запити ми просто ігноруємо */
        }
        return 0;
    }
}
int main (void)
{
    DDRB = 0b00001110; // PB1, PB2, PB3 - вихід
    usbInit ();
    usbDeviceDisconnect (); // Примусово відключаємося від хоста, так
робити можна тільки при виключених переривання!
    uchar i = 0;
    while (- i) {// пауза > 250 ms
        _delay_ms (1);
    }
    usbDeviceConnect (); // Підключаємося
    sei (); // Дозволяємо переривання
    for (;;) {// головний цикл програми
        usbPoll (); // Цю функцію треба регулярно викликати з головного
циклу, максимальна затримка між викликами - 50 ms
    }
    return 0;
}

```