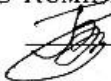


Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки


МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Апаратно-програмна система контролю та інформування учасників
колективної роботи»

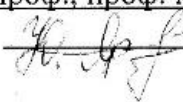
Виконав: магістрант 2 курсу, групи КІ-21 мз
спеціальності 123 Комп'ютерна інженерія

 Годун. В. В.

Керівник к.т.н., проф., проф. каф. ОТ

 Азарова А. О.

Опонент д.т.н. проф., проф. каф. МБІС

 Яремчук Ю. Є.

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

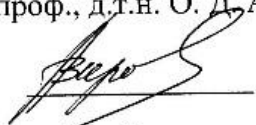
« 12 » 06 2023 р.

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О. Д. Азаров


«20» 03 2023 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Годуну Владиславу Володимировичу

1 Тема роботи «Апаратно-програмна система контролю та інформування учасників колективної роботи», керівник роботи Азарова Анжеліка Олексіївна к.т.н., професор,

затверджені наказом вищого навчального закладу від 20.03.2023 року № 68

2 Строк подання студентом роботи 9.06.2023 р.

3 Вихідні дані до роботи: процесор Core i7 5930K, Відеокарта не гірша за Radeon RX 480 G1, ОЗП не менше 32 ГБ із тактовою частотою 3200 МГц і вище, SSD накопичувач.


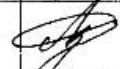


4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд та аналіз сучасного стану питання, розробка апаратної частини системи, розробка програмного забезпечення.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): електрична схема апаратної частини системи; список електронних компонентів;

команди налаштування GSM-модуля; лістинг програми надсилання повідомлень; лістинг програми обробки зворотного зв'язку; лістинг програми контролера системи

6 Консультантів розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Азарова А. О., к.т.н., професор		
4	Нікіфорова Л.О., к.е.н., доцент		

7 Дата видачі завдання 07.09.2021 р.

8 Календарний план наведено в табл. 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Аналітичний огляд предметної області	21.03.21	<i>best</i>
2	Аналіз способів контролю та інформування	22.03-02.04.23	<i>best</i>
3	Аналіз вимог	10.04-08.04.23	<i>best</i>
4	Розробка апаратної частини системи	09.04-29.04.23	<i>best</i>
5	Розробка ПЗ	30.04-19.05.23	<i>best</i>
6	Тестування системи	20.05-25.05.23	<i>best</i>
7	Розрахунок економічної частини роботи	26.05-.31.23	<i>best</i>
8	Оформлення пояснювальної записки та ілюстративного матеріалу	01.06-04.06.2.23	<i>best</i>
9	Аналіз виконання роботи, висновки, додатки	05.06-06.06.23	<i>best</i>
10	Перевірка якості виконання магістерської роботи та усунення недоліків	08.06.23	<i>best</i>

Студент



Годун В. В.

Керівник роботи



Азарова А. О.

АНОТАЦІЯ

УДК 004.93

Годун В.В. Апаратно-програмна система контролю та інформування учасників колективної роботи. Магістерська кваліфікаційна робота зі спеціальності 123 – компютерна інженерія, освітня програма комп'ютерна інженерія. Вінниця, ВНТУ 2023, 106 с.

Укр.мовою. Бібліогр.: 19 назв., рис. 30, табл. 14.

Дана магістерська кваліфікаційна робота присвячена проектуванню апаратно-програмної системи контролю та інформування учасників колективної роботи.

У роботі здійснено огляд та аналіз сучасного стану досліджуваного питання та проаналізовано способи контролю та інформування учасників колективної роботи. Розроблено апаратну частину системи та програмне забезпечення протоколу взаємодії апаратної частини з комп'ютером; створено програми контролера системи; а також запрограмовано функції клієнтського ПЗ. Проведено моделювання та тестування системи. Було розроблено інструкцію з використання запропонованого апаратно-програмного комплексу.

Ключові слова: інформування, система, апаратна частина, мікропроцесорна платформа, GSM мережі та зв'язок, колективна робота.

ANNOTATION

Godun V.V. The hardware and software system for monitoring and informing participants of collective work. Master's qualification route in the specialty 123 — computer engineering, educational program computer engineering. Vinnitsa, VTNU, 2025, 106 p.

In the Ukr. leng. Libr. name 19, figure 30, table 14.

This master's thesis is devoted to the design of a hardware and software system for controlling and informing participants of collective work.

The work includes a review and analysis of the current state of the issue and analyzes methods of control and informing. The hardware part of the system and the software of the protocol for the interaction of the hardware part with the computer have been developed; system controller programs; as well as programming of client software functions. Modeling and testing of the system was carried out. The instruction for use of the development was created.

Key words: information, system, hardware, microprocessor platform, GSM networks and communication, collective work.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД ТА АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ	
1.1 Аналітичний огляд предметної області	10
1.2 Актуальність роботи	12
1.3 Аналіз способів контролю та інформування	14
1.3.1 Месенджери, соціальні мережі, електронна пошта	14
1.3.2 Голосові роботи (скрипти)	14
1.3.3 Система коротких повідомлень (SMS)	15
1.4 Реінжиніринг процесу інформування клієнтів	15
1.5 Аналіз вимог	17
2 РОЗРОБЛЕННЯ АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ	
2.1 Опис функціональних зв'язків блоків	21
2.2 Організація живлення системи	26
2.3 Підключення GSM-модуля до контролера	29
2.4 Підключення плеєра	30
3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	
3.1 Розробка протоколу взаємодії апаратної частини з комп'ютером	32
3.1.1 Команди налаштування GSM-модуля	32
3.1.2 Команди, що передаються клієнтському ПЗ	32
3.2 Розробка програми контролера системи	33
3.3 Програмування функцій клієнтського програмного забезпечення	38
3.3.1 Загальні відомості	38
3.3.2 Перевірка формату номера телефону на коректність	38
3.3.3 Надсилання україномовних SMS повідомлень	40
3.3.4 Застосування кодування UCS2	41
4 ТЕСТУВАННЯ СИСТЕМИ	

08-23.МКР.002.00.000 ПЗ

Змн.	Лист	№ докум.	Підпис	Дата				
					Апаратно-програмна система контролю та інформування учасників колективної роботи Пояснювальна записка	Літ.	Арк.	Аркушів
							6	
						ВНТУ, гр. КІ-21мз		
<i>Розробив</i>		<i>Годун В.</i>						
<i>Керівник</i>		<i>Азарова А.О.</i>						
<i>Рецензент</i>		<i>Яремчук Ю.Є.</i>						
<i>Н. Контроль</i>		<i>Швець С.І.</i>						
<i>Затверджую</i>		<i>Азаров О.Д.</i>						

4.1 Математична модель системи.....	44
4.1.1 Каскадування пристроїв	45
4.1.2 Тестування моделі каскаду з двох пристроїв	46
4.1.3 Результати тестування	47
4.2 Інструкція користувача	47
4.2.1 Загальні відомості	47
4.2.2 Підключення апаратної частини.....	47
4.2.3 Режим розсилки.....	49
4.2.4 Підготовка списку розсилки	50
4.2.5 Запуск розсилки.....	50
4.3 Режим отримання зворотного зв'язку	51
4.3.1 Включення чергового режиму	51
4.3.2 Аналіз зворотного зв'язку	53
4.4 Системні вимоги	54

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки	55
5.2 Прогнозування витрат на виконання науково-дослідної роботи.....	58
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	63

ВИСНОВКИ	69
-----------------------	----

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	71
--------------------------------------	----

Додаток А Технічне завдання	73
--	----

Додаток Б Електрична схема апаратної частини системи	77
---	----

Додаток В Список електронних компонентів.....	78
--	----

Додаток Г Команди налаштування GSM-модуля.....	79
---	----

Додаток Д Лістинг програми відправки повідомлень	84
---	----

Додаток Е Лістинг програми обробки зворотного зв'язку.....	94
---	----

Додаток Є Лістинг програми контролера системи.....	101
---	-----

Додаток Ж Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	106
---	-----

						08-23.МКР.002.00.000 ПЗ	Арк.
							7
Змн.	Арк.	№ докум.	Підпис	Дата			

ВСТУП

У зв'язку з економічною кризою, що настала в Україні, організації все більше часу і сил витрачають на оптимізації власних бізнес-процесів. Як правило організації проводячи оптимізації ставлять собі за мету домогтися максимально можливого скорочення витрат. По суті завдання, що стоять перед організацією, зводиться до отримання колишніх результатів, але з меншими витратами. Особливу увагу організації звертають на інформаційні технології. Така увага зумовлена можливістю перекласти виконання різного роду завдань з людини на автоматизовану систему, обслуговування якої обходиться організації дешевше, ніж утримання співробітника, який виконує покладені на нього обов'язки. Сучасні технології дозволяють це робити, а тому створення такого роду систем є **актуальним**.

Мета роботи — покращення колективної роботи учасників за рахунок створення відповідної апаратно-програмної системи контролю та інформування.

Задачі дослідження:

- провести огляд та аналіз сучасного стану питання;
- проаналізувати способи контролю та інформування;
- розробити апаратну частини системи;
- реалізувати створену систему програмно;
- розробити протокол взаємодії апаратної частини з комп'ютером та програму контролера системи;
- реалізувати програмно функції клієнтського ПЗ;
- здійснити тестування системи;
- розробити керівництво користувача.

Наукова новизна отриманих результатів магістерської роботи полягає в удосконаленні методу масового інформування осіб, які задіяні у діяльності організації, шляхом реалізації процесу автоматичного отримання зворотного зв'язку, що дозволяє скоротити використання людських ресурсів під час

виконання підтримуючого процесу оброблення заявок та уможливило скорочення витрат за рахунок впровадження засобів автоматизації.

Об'єкт дослідження — процес проектування апаратно-програмних систем на базі мікропроцесорних платформ та модулів GSM зв'язку.

Предмет дослідження методи розробки апаратно-програмних систем з використанням мікропроцесорних платформ та модулів GSM зв'язку.

Публікація результатів магістерської роботи здійснювалася шляхом написання двох авторських свідоцтв на створений апаратно-програмний комплекс.

Апробація результатів магістерської роботи: подано до публікації 2 авторських свідоцтва на розроблений програмно-апаратний комплекс.

1 ОГЛЯД ТА АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ

1.1 Огляд предметної галузі

В адміністрації будь-якого підприємства часто виникають ситуації, коли необхідно за максимально короткий термін сповістити велику кількість людей. Передається інформація, як правило, має високу важливість, тому слід враховувати специфіку та надійність каналу, яким передбачається інформування. Також важливу або навіть вирішальну роль може відігравати швидкість доставки повідомлень. Недооцінивши ці фактори, можна надіслати людині термінове повідомлення до месенджера, у якого заблоковані оповіщення, або надіслати у сповіщення в соціальній мережі доступ до якої людина звикла здійснювати з домашнього комп'ютера. Дорогоцінний час буде втрачено. Проблема стає актуальною для організацій, чия діяльність залежить від миттєвої реакції суб'єкта на інформаційний вплив з боку організації.

Найбільш надійним варіантом інформування є простий дзвінок по телефону, оскільки зворотний зв'язок від людини надходить миттєво, під час сеансу зв'язку. При цьому сповіщення по телефону кожної людини забирає значний час і потребує безпосередньої участі секретаря у цьому процесі.

Ситуація посилюється тим, що далеко не кожен із потенційних клієнтів готовий негайно ухвалити рішення. Клієнту треба подумати, чи зможе він взяти участь у роботі, і в одну-дві години приїхати на місце проведення запланованих заходів.

Усіх оповіщених клієнтів можна умовно поділити на групи:

- клієнти, які підтвердили свою участь;
- клієнти, які не можуть брати участь зараз, але, як і раніше, зацікавлені в отриманні інформації про майбутні заходи;
- клієнти, які втратили інтереси наступне інформування недоцільно;
- клієнти, статус яких не визначено (недоступний, не бере слухавку..

Заздалегідь невідомо хто з клієнтів поставиться до якоїсь групи. У зв'язку з цим, телефонувати доводиться до всіх потенційних клієнтів номери телефонів,

яких доступні секретареві. Оскільки всі операції з інформування виконуються вручну, можна говорити про повну відсутність автоматизації на рівні управління комунікаціями.

Керуючись цими фактами, прийнято рішення створити систему інформування за допомогою якої організація зможе забезпечити надійне інформування людей і отримання від них зворотного зв'язку, і при цьому обходитися без штатного секретаря.

Передбачається, що цільова система буде програмно-апаратним комплексом, за допомогою якого будь-який співробітник зможе масово інформувати зацікавлених осіб і отримувати від них зворотний зв'язок. При цьому в процесі безпосереднього інформування участь співробітника по можливості слід мінімізувати.

На сьогоднішній момент не існує універсальної моделі інформування клієнтів, яку можна було б знайти у вільних джерелах. Зрозуміло, у кожній організації є своя бізнес-модель, але це не означає, що вона універсальна або вони її готові надати широкому загалу. У зв'язку з цим кожна нова організація змушена створювати та моделювати свої процеси.

У цій роботі передбачається створення типової та самодостатньої моделі, яка:

- враховуватиме вузьку специфіку організації;
- може застосовуватись у чистому вигляді (як є, без зміни);
- послужить відправною точкою при початковому моделюванні бізнес-процесів у нових організаціях зі схожою специфікою.

Скориставшись такою моделлю будь-яка організація зможе застосовувати її з урахуванням власної специфіки, і при цьому не витратить значних ресурсів на початкове вивчення та моделювання супутніх процесів. Наявність такої моделі заощадить багато сил і коштів.

Для створення такої системи необхідно формалізувати предметну область у вигляді моделі, та запропонувати рішення щодо автоматизації описаних у даній моделі бізнес-процесів. У рамках магістерської роботи автор спробував

створити таку систему.

1.2 Актуальність роботи

Актуальність застосування можна обґрунтувати, оцінивши користь, яку отримує організація під час застосування аналізованої системи. Результати застосування слід розділити на якісні та кількісні.

Якісні:

- підвищення ефективності управління комунікаціями;
- більш ефективне виконання робочих операцій;
- підвищення загальної результативності роботи за рахунок більш раціональної її організації.

Кількісні:

- зниження постійних витрат;
- зниження операційних витрат.

Зниження постійних витрат відбувається за рахунок відсутності необхідності наймати співробітника для управління комунікаціями та виплачувати йому заробітну плату. Зниження операційних витрат відбувається з допомогою економії з оплати послуг зв'язку.

Для прикладу, можна спробувати оцінити, скільки організація не заробляє (втрачає) поки що для інформування клієнтів вона змушена наймати секретаря, який у процесі виконання своїх обов'язків витрачає кошти організації на здійснення вихідних телефонних дзвінків. Ці обставини змушують запровадити інструмент, який дозволить забезпечувати комунікації із зовнішнім світом, за умов гострого дефіциту чи повної відсутності вільних людських ресурсів у організації. У цій роботі розроблено процес комунікації, який вимагає безпосередньої участі людини. Користувач під час роботи з такою системою спочатку виступає у ролі ініціатора. Задавши деякі вступні дані, він запускає інформатор. Надалі користувач виступає у ролі аналітика, коли досліджує звіт

про зворотний зв'язок.

Дії співробітника, який відповідає за інформування клієнтів, слід розглянути докладніше.

Якщо припустити, що на розмову з одним клієнтом піде 3 хвилини, то на дзвінки 20 клієнтам секретар витратить 1 годину часу.

Секретар не може розраховувати на те, що всі, кому він зателефонує, негайно підтримають запропонований порядок денний. Перш ніж клієнт зможе прийняти об'єктивне рішення, йому як мінімум необхідно:

- бути готовим отримати повістку у запропонованому секретарем формі;
- ознайомитись із повісткою;
- відреагувати на порядок денний (дати зворотний зв'язок).

У процесі інформування клієнти діляться наступним ознаками:

— клієнт не надав зворотний зв'язок (не отримав інформацію і не має можливості на неї зреагувати), ця група неспроможна представляти інтересу організації;

— клієнт надав зворотний зв'язок. У цьому випадку клієнти групуються за типом отриманого зворотного зв'язку;

— клієнт зацікавлений у порядку денному;

— клієнт не зацікавлений у порядку денному, але зацікавлений у отриманні інформації про майбутні заходи;

— клієнт відмовився від подальшого інформування.

З точки зору тимчасових витрат на інформування між цими групами немає жодної різниці, оскільки секретареві на етапі угруповання доводиться дзвонити кожному з них.

Як показує практика, секретар може обдзвонити порядком 20-30 номерів на годину. При прогнозі конверсії від дзвінків, що дорівнює 0,1 (що є досить оптимістичним прогнозом), в результаті роботи секретаря можемо отримати 2/3 клієнта на годину, що явно недостатньо. Економічно доцільний норматив передбачає обслуговування щонайменше 20 клієнтів на годину.

1.3 Аналіз способів інформування

1.3.1 Месенджери, соціальні мережі, електронна пошта

Рішення відмовитися від інформування через месенджери та соціальні мережі пов'язане із вимогами до надійності інформування. Усі представлені способи інформування мають одну загальну властивість. Для роботи всіх зазначених засобів зв'язку потрібна наявність Інтернету. Для більшості клієнтів отримання повідомлень не складе жодної проблеми, оскільки широкопasmовий доступ до мережі доступний більшості людей, що фізично перебувають у межах міста. У той же час, для управління такого роду комунікаціями, доступом до мережі повинна мати і організація. Оскільки експлуатація може відбуватися у віддалених районах, на стабільний та якісний зв'язок розраховувати не доводиться. Як правило, при виїзді з міста якість зв'язку помітно падає. Якщо використати такий канал інформування, нам слід зважати на вимогу до надійності інформування з боку організації, що неможливо без стабільного доступу до мережі інтернет. Висновок: в умовах роботи у віддалених районах застосування даного способу інформування може бути утрудненим або неможливим.

1.3.2 Голосові роботи (скрипти)

Теоретично функцію дзвінків клієнтам можна доручити роботу, яка автоматично здійснить виклик кожного клієнта та озвучить повістку, проте є одна важлива обставина. Останнім часом дзвінки роботів сприймаються не інакше, як спам. Як наслідок номер робота може з часом потрапити до спаму листів операторів.

У той же час, якщо клієнт сам дзвонить в організацію з будь-якого питання і йому відповідь голосовий робот. Це буде сприйнято як щось зрозуміле.

Резюмуючи дану обставину було сформовано вимогу: Система може

обслуговувати вхідні дзвінки, але не може здійснювати вихідні.

1.3.3 Система коротких повідомлень (SMS)

Для абонентів GSM мереж SMS повідомлення є найбільш доступним способом інформування. З 1992 року отримання та відправлення SMS повідомлень стали стандартом для абонентів GSM-мереж. SMS зарекомендували себе як надійний спосіб інформування. Можна відзначити, що й досі найважливіші та термінові повідомлення (попередження МНС, авторизація, банкінг) передаються у вигляді SMS.

Надійність та простота даного способу інформування обумовлена структурою системи пакетної передачі даних у мережах GSM. Свого часу, при проектуванні протоколу, інженери, що стояли біля витоків GSM, розробили структуру пакета даних. Було ухвалено рішення всередині кожного пакета, без певної на той момент мети, зарезервувати 140 байт даних. Їхні послідовники вирішили навантажити цей абсолютно марний, на той момент трафік, корисною функціональністю. Теоретичні міркування аналітиків зводилися до того, що для абонента GSM мережі немає жодної різниці, чи прийдуть йому 140 байт нулів або він отримає якусь корисну інформацію. У практичному сенсі, абонент у кожному пакеті може отримувати 160 символів у 1-байтному кодуванні або 70 символів у 2-байтному кодуванні. При цьому апарат абонента має справу з низькорівневим трафіком,

У загальному випадку абонент GSM мережі отримає SMS повідомлення навіть при негативному балансі особового рахунку. У найбільш розважливих операторів ця можливість може бути примусово заблокована, проте це досить рідкісне явище.

Всі перелічені обставини дозволяють розглядати SMS повідомлення як надійний спосіб інформування, який поступається за надійністю лише прямим телефонним дзвінкам.

1.4 Реінжиніринг процесу інформування клієнтів

Як канал зв'язку вибрано GSM. Телефон є найбільш консервативним та очікуваним засобом зв'язку. При комунікації з клієнтом цю обставину слід враховувати. Важко уявити «серйозну» організацію без телефону. Без телефону наймовірно миттєве встановлення надійного зв'язку між клієнтом та організацією хто б не був ініціатором комунікації. У той же час, GSM завдяки SMS дозволяє надійно інформувати клієнтів, не вдаючись до телефонних дзвінків.

Під час проведення реінжинірингу було висунуто гіпотеза у тому, що немає необхідності обдзвонювати всіх клієнтів. Достатньо надіслати інформацію у вигляді SMS повідомлень. Оскільки наявність клієнтської бази передбачає, що клієнти самі залишили свій телефон, надходження повідомлень буде очікуваною для них подією. У подальших міркуваннях передбачається, що клієнти так чи інакше чекають на надходження інформації.

Клієнт отримає SMS-повідомлення. Якщо клієнт зацікавлений в отриманому порядку, йому не важко зателефонувати в організацію. Якщо зараз порядок денний для клієнта не актуальний, не потрібно робити нічого. Достатньо просто проігнорувати отримане повідомлення. Таким чином секретар не витрачає час на розмови з не зацікавленими у порядку клієнтами, а клієнти у свою чергу позбавлені необхідності відволікатися на вхідні телефонні дзвінки.

Таким чином інформування клієнта здійснюється через передачу SMS повідомлення з повісткою, а обробка зворотного зв'язку здійснюється за допомогою прийому вхідних телефонних дзвінків від клієнтів, що виявили інтерес до повістки.

Після реінжинірингу бізнес-процесів організації завдання секретаря звелися до ініціації інформування та перегляду звіту системи. Після вивчення звіту секретар може швидко ухвалити рішення з ким із клієнтів слід зв'язатися. У такому разі секретар дзвонитиме лише цільовим клієнтам. Враховуючи, що

система перебирає комунікації з клієнтом, можна говорити про відсутність необхідності призначати окремо людини на роль секретаря. Обслуговування системи може здійснювати будь-який співробітник організації.

Процес обробки зворотного зв'язку передбачає прийом телефонних дзвінків. Як правило, кожному клієнту доводиться доносити абсолютно однакову інформацію. Це інформація про час проведення зустрічі, місце зустрічі чи інша інформація необхідна кожному клієнту. Враховуючи дані обставини, у секретаря немає необхідності постійно повторювати цю інформацію. Теоретично, достатньо запрограмувати робота, який озвучуватиме повістку автоматично, без участі секретаря. Для реалізації описаних функцій складено технічне завдання на розробку засобів автоматизації.

Слід зазначити, що отримання зворотного зв'язку під час роботи системи перестало бути обов'язковим. Інформаційна розсилка може мати повідомний характер, яка вимагає дій від одержувача.

1.5 Аналіз вимог

Якщо зібрати докупи інформацію, отриману від представників організацій на етапі передпроектного дослідження, отримуємо такі дані:

- існує база із номерами телефонів потенційних клієнтів;
- актуальне зведення співробітники організації отримують за 2-3 години на початок зустрічі, відповідно система, що розробляється, повинна бути розрахована на використання в умовах крайнього дефіциту часу та людських ресурсів;
- якщо отримано сприятливе звернення, приймається рішення сповістити клієнтів і запросити їх на зустріч;

Головні вимоги, організацією до такої системи:

- надійність інформування;
- мінімальна участь співробітників організації у роботі системи;

- повна конфіденційність клієнтської бази;
- прив'язка до єдиного номера телефону;
- можливість збільшити швидкість інформування.

Усі отримані вимоги до системи, що розробляється, визначено, а на їх підставі було складено ТЗ. Повне ТЗ представлено в додатку А. Всі вимоги умовно поділені на вимоги до апаратних функцій та вимоги до програмних функцій. У табл. 1.1 наведено перелік найбільш значущих вимог до апаратних функцій. У табл. 1.2 наведено перелік найбільш значущих вимог до програмних функцій.

Таблиця 1.1 — Вимоги до апаратних функцій

Код	Вимога	Примітка
F.01.00	Загальні вимоги	
F.01.01	Можливість надсилати SMS повідомлення через GSM мережу	V.01.01 BP.01.02
F.01.02	Можливість автоматичної відповіді на вхідний дзвінок	V.01.02
F.01.03	Можливість передавати інформацію про натиснуті під час розмови клавішах	F.02.04
F.01.04	Можливість передачі інформації про тривалості сеансу зв'язку	
F.01.04	Наявність USB інтерфейсу обміну даними з ПК	
F.01.05	Наявність протоколу обміну даними з ПК	Для отримання параметрів SMS
F.01.06	Можливість отримувати параметри розсилки від ПК	
F.01.06.01	Номер телефону абонента	F.01.06
F.01.06.02	Повідомлення	F.01.06
F.01.07	Програвання повідомлень при вхідному дзвінку	F.01.02

Таблиця 1.2 — Вимоги до програмних функцій

Код	Вимога	Примітка
F.02.00	Загальні вимоги	
F.02.01	Можливість надсилання SMS повідомлення кільком абонентам	Групове розсилання Однакового повідомлення всім користувачам
F.02.02	Можливість надсилання SMS за шаблоном	F.02.05 Користувач готує файл із розсилкою, програма читає та доповнює інформацією наявний файл
F.02.03	Можливість надсилання SMS повідомлення одному абоненту	Персоналізоване надсилання унікальних повідомлень кожному користувачеві
F.02.04	Програма повинна мати можливість отримувати інформацію про натиснуті час розмови клавішах	F.01.03
F.02.05	Можливість використовувати файли в форматі xlsx як шаблон для розсилок	I.02.03.01
F.02.06	Підтримка україномовних SMS	
F.02.06.01	Програма повинна мати можливість налаштування порту модему	I.02.02.03F.01.04 для роботи з GSM мережею для відправки SMS та отримання зворотного зв'язку в системі має бути встановлено модем
F.02.06.02	Пошук доступних COM портів	
F.02.06.03	Вибір номера порту	
F.02.06.04	Вибір швидкості порту	
F.02.06.05	Автоматичний пошук модему	Надіслати AT
F.02.07	Налаштування списку розсилки	
F.02.07.01	Вибір файлу із шаблоном розсилки	I.02.03.01

F.02.07.02	Вставка списку розсилки з буфера обміну	
F.02.08	Генерація файлу звіту про роботу системи	
F.02.09	Вибір періоду звіту про роботу системи	
F.02.10	Розрахунок кількості повідомлень при підготовці розсилки	
F.02.11	Можливість отримання зворотного зв'язку при організації розсилки	
F.02.12	SMS оповіщення диспетчера про запит абонентом термінового зворотного зв'язку	

2 РОЗРОБЛЕННЯ АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ

2.1 Опис зв'язків функціональних блоків

Спочатку слід визначитися з термінами визначеннями та поняттями. У цьому роботі GSM модулем чи GSM модемом називатиметься модуль SIM900, вироблений компанією Simcom [4]. Плеєром називатиметься модуль DFPlayermini. Як контролер системи застосовується плата ArduinoNano v3 [6].

Схема взаємодії блоків системи, що застосовується в даній роботі, представлена на рис. 2.1.

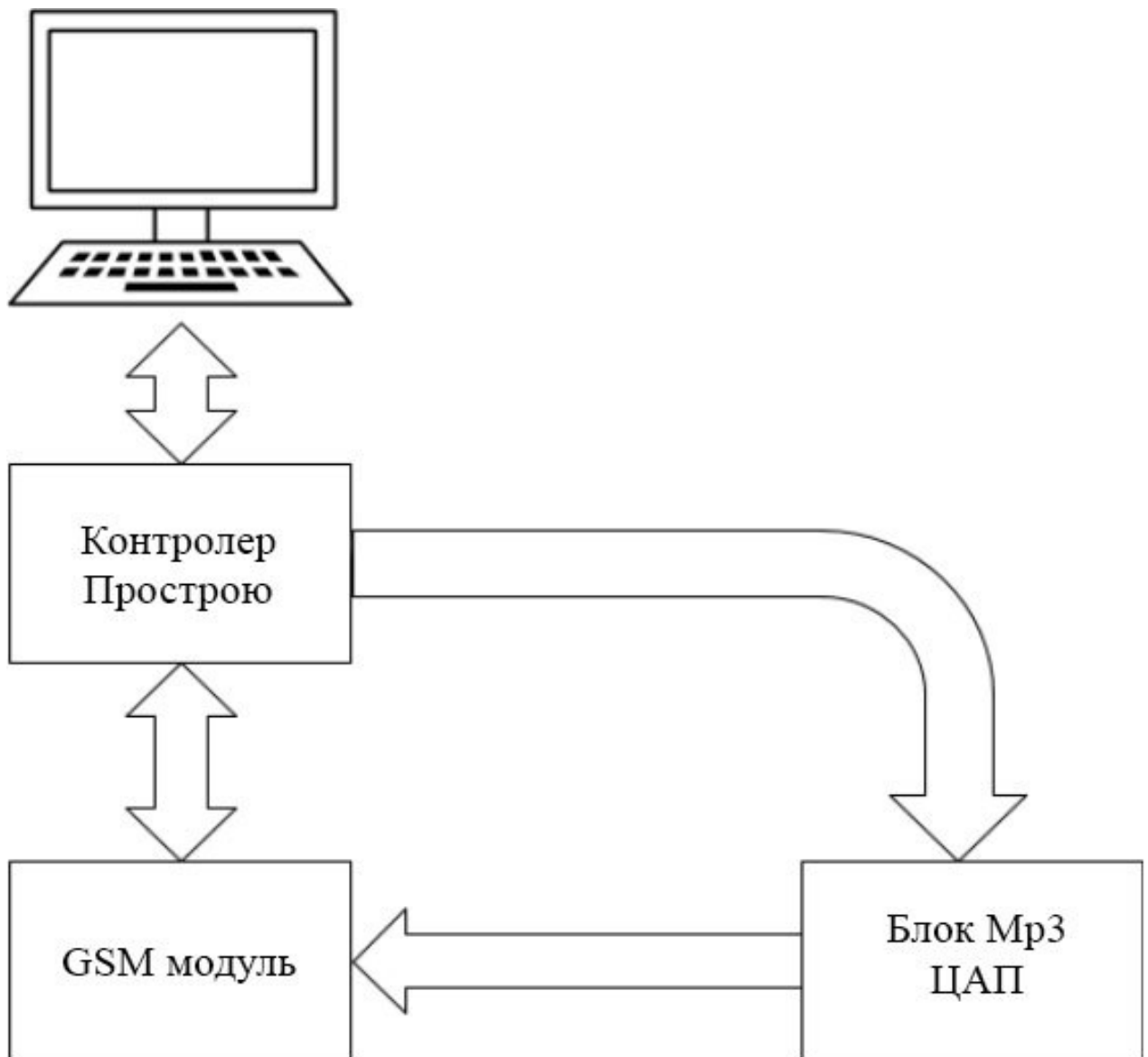


Рисунок 2.1 — Схема взаємодії блоків системи

Згідно з представленою схемою необхідно забезпечити наступні зв'язки:

- зв'язок ПК із контролером;
- зв'язок контролера із GSM-модулем;
- зв'язок контролера з плеєром (блок mp3 ЦАП);
- зв'язок програвача з GSM-модулем.

Проаналізувавши різні мікропроцесорні платформи на основі Arduino було прийнято рішення вибрати Arduino Nano (рис. 2.2).

Arduino Nano — це маленький, але повнофункціональний пристрій. Він побудований на основі мікроконтролера ATmega328. Також він схожий на Arduino Duemilanove. Різниця між ними — це відсутність роз'єму для живлення, розмір та типом USB-кабелю. ATmega168 і ATmega328 випускається з прошитим завантажувачем. Автоматичне (програмне) скидання відбувається за допомогою програми з підключеного комп'ютера. Джерела живлення 5В.



Рисунок 2.2 — Контролер Arduino Nano

Тепер розглянемо її більш детально, а саме: переваги, робота з пам'яттю, її живлення та коротку характеристику.

Основні переваги:

- компактність: вона є найменшою з усіх Arduino, це дозволяє виготовляти компактні пристрої;
- ціна: вона є недорогою порівняно з іншими, тож для недосвідчених людей це буде хорошим варіантом спробувати зробити пристрій;
- зрозумілість: ця плата має зручний та доступний інтерфейс, що дозволяє швидко розпочати роботу;
- відкритий код: тобто користувач має доступ до великої кількості бібліотек та програмного забезпечення;

Плата має низькі вимоги до живлення та буде працювати при напрузі від 7 до 12В. Також перевагою є те, що вона працює від різних джерел живлення: батареї, блок живлення та інші.

Arduino Nano має 32 кілобайт програмної пам'яті та 2 кілобайти ОЗП. Цього достатньо для простих проектів, але для складніших проектів може виникнути необхідність у більшій кількості пам'яті. У таких випадках можна використовувати зовнішню пам'ять, таку як EEPROM або SD-карту, щоб зберігати дані та програмне забезпечення.

Обрана плата, як і більшість мікроконтролерів, може бути автономним, якщо він живиться від батареї або іншого джерела живлення з високою ємністю, таким як сонячна батарея або пальчикові батареї. Однак, тривалість роботи в автономному режимі залежить від споживання енергії пристроєм, його режиму роботи і використання додаткових модулів.

Для досягнення максимальної автономності можна використовувати різні техніки, такі як режим сну, коли мікроконтролер переходить в режим низького споживання енергії, підключення датчиків руху або освітлення, що дозволяє вмикати пристрій тільки при виявленні руху або зміні освітлення в приміщенні, а також оптимізація програмного коду.

Мікроконтролер ATmega328P, на якому базується Arduino Nano, є високопродуктивним, низькопотужним мікроконтролером, що має широкі можливості для використання в різних проектах. Цей мікроконтролер

розроблений компанією Atmel (тепер Microchip Technology) та має такі основні технічні характеристики:

- архітектура RISC (Reduced Instruction Set Computing) з 8-бітним та 16-бітним наборами інструкцій;
- частота роботи від 1 до 20 МГц з можливістю подальшого розгону до 20 МГц;
- 32 кБ флеш-пам'яті для зберігання програмного коду(на цю пам'ять записується скетч, який виконується при запуску мікроконтролера. Є можливість зберігати кілька скетчів);
- 2 кБ ОЗУ для зберігання змінних та стеку виклику(дозволяє зберігати дані під час виконання програмного коду та передавати їх для обробки);
- 1 кБ EEPROM для зберігання даних, які мають зберігатись між запусками мікроконтролеру;
- вбудований 10-бітний ADC (аналогово-цифровий конвертер) з 6 входами для зчитування аналогових сигналів.
- вбудований 6-канальний ШІМ (Широтно-імпульсна модуляція) для керування сервоприводами та іншими електромеханічними пристроями;
- Вбудовані модулі UART, SPI та I2C для забезпечення зв'язку з різними зовнішніми пристроями.

ADC (Analog-to-Digital Converter) у мікроконтролері ATmega328P дозволяє зчитувати аналогові сигнали з датчиків та перетворювати їх на цифрові значення. Це важлива функція, яка дозволяє контролювати фізичні величини.

Arduino Nano має 6 входних пінів для зчитування аналогових сигналів, позначених як A0-A5. Кожен з цих пінів може зчитувати значення від 0 до 5 вольт, що відповідає діапазону від 0 до 1023, де 0 – це 0 вольт, а 1023 – це 5 вольт. ADC у ATmega328P використовує методику приростового перетворення, яка дозволяє зчитувати значення з високою точністю та швидкістю. Кожен входний пін, підключений до датчика, має вбудований опір для збалансування входного струму.

ШІМ (PWM) – це технологія, що дозволяє керувати виходом сигналу з

плавною зміною його ширини і частоти. ШІМ може бути використаний для керування швидкістю обертання моторів, рівнем яскравості світлодіодів, керування сервоприводами та іншими електромеханічними пристроями, що потребують плавного керування.

Крім того, наявність каналів ШІМ дозволяє створювати складніші проекти, що включають керування багатьма електромеханічними пристроями, які потребують плавного керування. Завдяки можливості керування до 6-ти пристроїв на одній платі Arduino Nano, зменшується кількість необхідних компонентів і складність збірки електронної системи.

Arduino Nano використовує мікроконтролер ATmega328P як основну частину плати, що дозволяє використовувати всі його можливості для розробки різноманітних проектів. Це дозволяє використовувати весь спектр функцій та можливостей цього мікроконтролера, щоб створювати різноманітні проекти з електронікою, автоматизації та інших сфер.

GSM-модуль, який застосовувався при створенні прототипу системи, був попередньо розпаяний на платі. Для промислового виконання можна використовувати модуль у вигляді чіпа та розпаяти його самостійно на попередньо розведений плату. Ядром модуля є контролер MediaTek ARM mt6261 [7]. Зв'язок у мережах GSM забезпечує 4 діапазонний приймач rf7198 – рис. 2.3.

Даний модуль обраний автором оскільки це один з найдоступніших модулів на ринку. При цьому модуль відпускається виробниками за відносно низькою ціною. Вартість модулів лінійки 900 коливається від 120 до 500 гривень.

Для роботи модуля необхідно вставити у відповідний слот робочу SIM-картку у форматі microSIM.

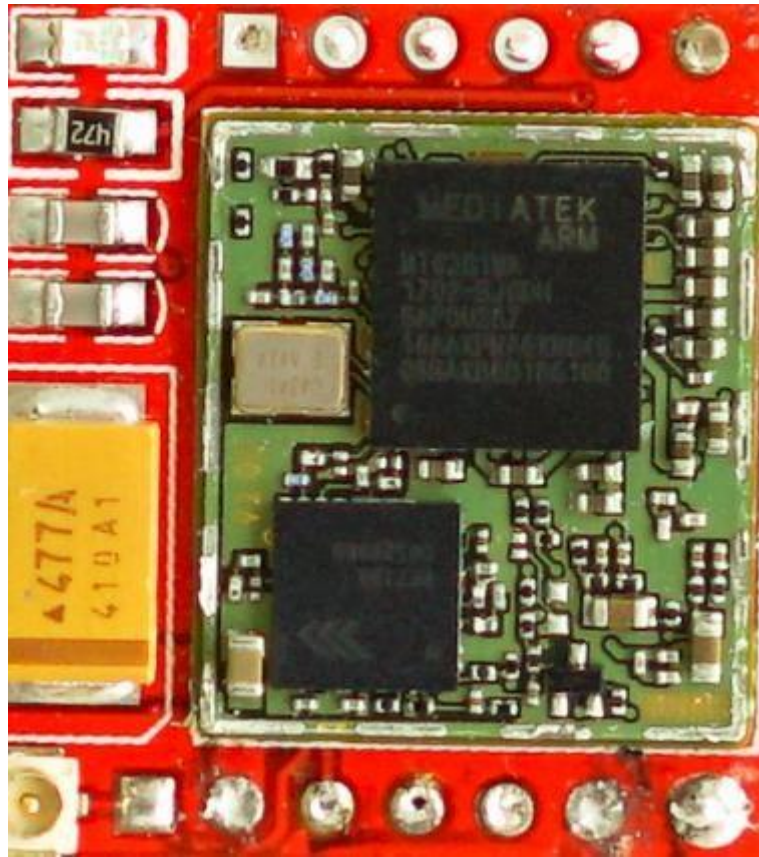


Рисунок 2.3 — Модуль SIM900

2.2 Організація живлення системи

GSM-модуль потребує напруги живлення в діапазоні від 3,4 до 4,5 вольт. Рекомендоване виробником живлення становить 4 В [4]. Це не найпоширеніша напруга живлення, тому при підключенні модуля необхідно подбати про джерело живлення. У робочому режимі модуль споживає 1.5 – 2 А струму що унеможливує живлення модуля безпосередньо від USB порту. Живлення від Arduino теж неможливо, оскільки Arduino неспроможна забезпечити модуль струмом вище 1А. При неправильному живленні пристрою можливі збої в його роботі.

При подачі живлення невідповідного вимогам у модулі передбачені механізми сигналізації та захисту. При подачі некоректної напруги модуль видасть попередження UNDER-VOLTAGE WARNING. У разі перевищення

порога модуль видасть повідомлення UNDER-VOLTAGE POWER DOWN і вимикається. Аналогічно відбувається при недостатній напрузі спочатку піде попередження OVER-VOLTAGE WARNING потім модуль видасть попередження OVER-VOLTAGE POWER DOWN і вимкнеться (рис. 2.4).

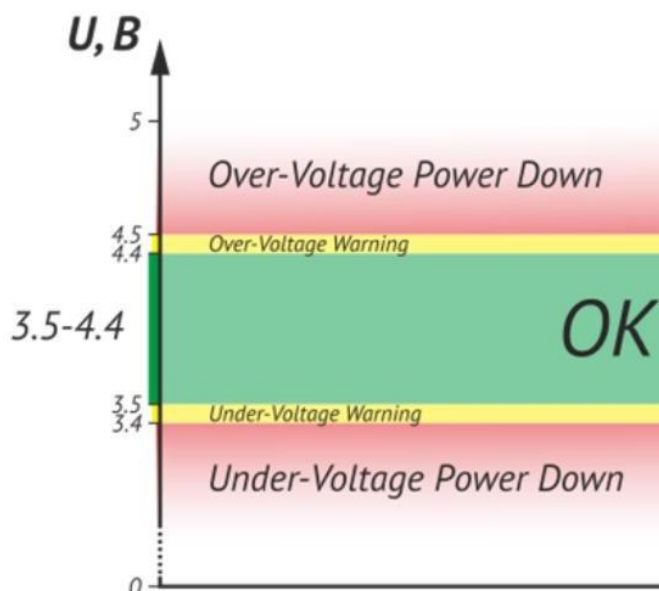


Рисунок 2.4 — Перевірка правильності живлення модуля SIM900

Для вирішення проблеми з живленням прийнято рішення живити модуль GSM від окремого джерела живлення, а контролер живити від USB порту комп'ютера. Це рішення обумовлено безумовною необхідністю підключати контролер до ПК передачі даних. При цьому всі пристрої повинні бути підключені до загальної землі. Для живлення застосований перетворювач LM2596 (рис. 2.5).

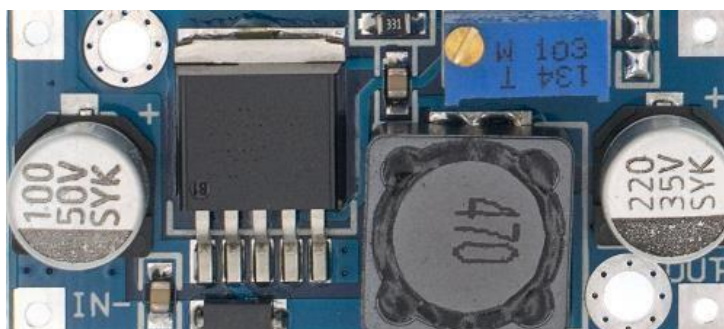


Рисунок 2.5 — Перетворювач напруги LM2596

Також у пристрої присутній блок живлення, який забезпечує стабільну постійну напругу та струм для різних електронних пристроїв. Він перетворює вхідну напругу з мережі в потрібну вихідну напругу, яка необхідна для правильної роботи приладу.

Блок живлення може бути внутрішнім або зовнішнім. Внутрішній блок живлення зазвичай встановлюється внутрішньо в електронному пристрої і підключається до живлення від мережі через кабель. Зовнішній блок живлення має свій власний корпус і підключається до електронного пристрою через кабель(його і будемо використовувати).

Основними параметрами блоків живлення є вихідна напруга та струм, які відповідають потребам конкретного електронного пристрою. Для нашого пристрою використовуємо напругу 5В. Також важливим параметром є ефективність блоку живлення, яка відображає відсоток електроенергії, яка перетворюється в корисну енергію, а не розсіюється у вигляді тепла.



Рисунок 2.6 — Зовнішній блок живлення

Після організації живлення модуль GSM необхідно підключити до пристрою управління. Як керуючий пристрій застосовується плата ArduinoNano.

2.3 Підключення GSM-модуля до контролера

При підключенні GSM-модуля до послідовного порту Arduino виникає проблема узгодження рівнів логічної одиниці.

Застосований у цій роботі GSM-модуль має стандартні рівні логічної одиниці і логічного нуля приєднання послідовного порту.

У документації до модуля вказаний діапазон допустимих рівнів логічної одиниці – рисунок 2.7 [4]. Рівень коливається від 2,1 до 3,1 вольт. Перевищення даних значень неминує веде до виходу з ладу модуля. Ця обставина вимагає особливої уваги з боку технічного фахівця, який здійснює розробку пристроїв на базі модулів серії SIM9xx[4].

Table 10: Serial port characteristics

Symbol	Min	Typ	Max	Unit
V _{IL}	-0.3	-	0.7	V
V _{IH}	2.1	-	3.1	V
V _{OL}	-	-	0.4	V
V _{OH}	2.4	2.8	-	V

Рисунок 2.7 — Рекомендовані рівні живлення SIM900

На початковому етапі проектування узгодження логічних рівнів модему і контролера передбачалося застосування спеціалізованих мікросхем перетворення рівнів. При докладнішому вивченні цього питання від цієї ідеї було вирішено відмовитися, оскільки потрібно узгодження лише однієї напівдуплексної лінії зв'язку. Набагато практичніше для вирішення цього завдання застосувати рівноплечий дільник напруги на двох резисторах. Дільник був встановлений на лінію послідовного порту контролера (RX). Теоретично для виготовлення дільника можна використовувати 2 однакові резистори номіналом від 1.5 до 10 кОм. Схему підключення дільника до контролера представлено на рис. 2.8.

Зміни рівня на лінії передачі порту (RX) GSM-модуля не потрібно оскільки контролер нормально сприймає рівень 2.8 В як логічну одиницю.

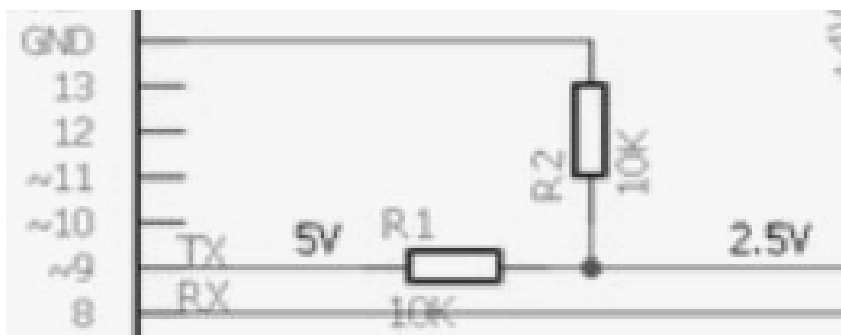


Рисунок 2.8 — Дільник напруги

2.5 Підключення плеєра

Єдине завдання MP3 плеєра в проєктованій системі — відтворення аудіофайлів для озвучення автовідповідача.

При отриманні тієї чи іншої команди від контролера системи програвач відтворює той чи інший файл формату mp3. У кожний час система може звернутися до будь-якого файлу на SD карті.

MP3 плеєр живиться від того ж джерела живлення від якого заживлений модем, оскільки згідно з характеристиками, заявленими виробником має схожі з модемом параметри робочої напруги. Напруга живлення плеєра 3,3-5 В. Плеєр працює зі стандартними (TTL) рівнями логічної одиниці на послідовному порту. При підключенні плеєра до контролера жодних додаткових погоджень не потрібно.

Аудіо файли для подальшого відтворення розміщуються в директорії mp3 на карті SD.

Модуль плеєра підтримує карти пам'яті формату microSD відформатовані у файлових системах FAT16 та FAT32.

Імена файлів повинні починатися з чотиризначних цифр. Інакше передбачуване управління відтворенням через послідовний порт неможливо.

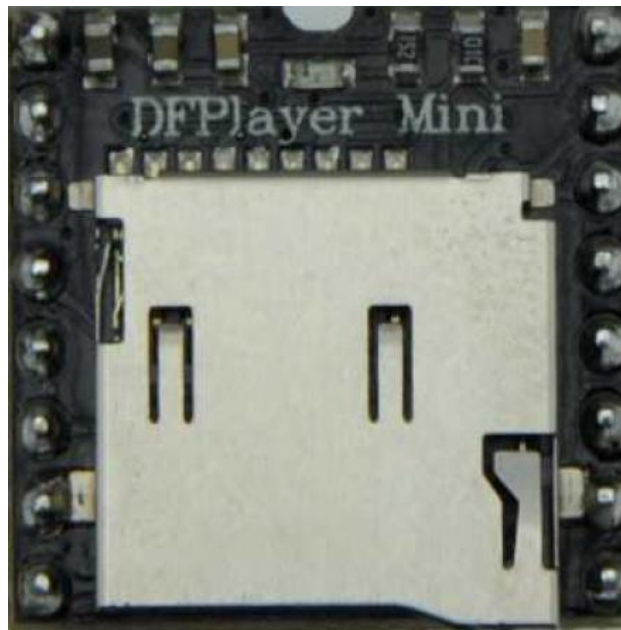


Рисунок 2.9 — Модуль DFPlayer mini

Робота плеєра керується контролером. Контролер відповідно до робочої програми передає команди в шістнадцятковому (hex) форматі на послідовний порт плеєра [5].

У даного модуля розпаяні вихідні лінії цифра аналогового перетворювача (ЦАП). Вихід ЦАП безпосередньо підключений до аналогового входу модуля GSM (MIC1, MIC2). Таким чином у момент, коли GSM – модуль обслуговує сеанс зв'язку, аудіо сигнали, що відтворюються плеєром, транслюються абоненту, з яким встановлено зв'язок.

У цьому роботі вибір амплітуди сигналу виконувався емпіричним шляхом. Оскільки аудіовхід GSM-модуля досить чутливий, амплітуда аудіо сигналу, що подається, склала лише 20% від номінальної амплітуди. Відповідно, значення за замовчуванням повинно бути зменшено в 3 рази. Цю обставину слід враховувати під час розробки програми контролера. Оскільки в плеєрі для підстроювання амплітуди вихідного аудіо сигналу на виході ЦАП передбачено низку відповідних команд.

3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка протоколу взаємодії апаратної частини з комп'ютером

Сукупність команд, застосовуваних системи, утворюють протокол взаємодії пристрою з ПК. Для зручності сприйняття всі команди умовно поділені на такі групи:

- команди налаштування модуля;
- команди для здійснення дзвінків та відправлення SMS;
- невідомі повідомлення від GSM-модему;
- Команди, що передаються клієнтському ПЗ.

3.1.1 Команди налаштування GSM-модуля

Команди налаштування застосовується для зміни поточної конфігурації модему, увімкнення/вимкнення функцій тощо.

Повідомлення, що не запитуються — це дані, які модем передає контролеру при настанні будь-яких подій або при зміні стану пристрою. Повідомлення, що не запитуються, може бути вхідний телефонний дзвінок, інформація про зміну стану GSM мережі, недостатня або надмірна напруга живлення модему і т.д. Детальний опис усіх необхідних команд представлено у додатку М.

3.1.2 Команди, що передаються клієнтському ПЗ

Команди, які клієнтське ПЗ може отримати від контролера системи представлені в таблиці 3.1. Передбачається, що всі інші повідомлення, що приходять від пристрою, що не запитуються, будуть ігноруватися. Всі команди, призначені для обробки в керуючу програму компютера, містять суфікс >.

Таблиця 3.1 — Команди, передані клієнтському ПЗ

Команда	Опис
>Tel: + number	Вхідний дзвінок із номера “number”
>NO *	Сеанс телефонного зв'язку закінчено
>Key: 0	Користувач натиснув клавішу 0
>Key: 1	Користувач натиснув клавішу 1
>Key: 2	Користувач натиснув клавішу 2
>Key: 3	Користувач натиснув клавішу 3
>Key: 4	Користувач натиснув клавішу 4
>Key: 5	Користувач натиснув клавішу 5
>Key: 6	Користувач натиснув клавішу 6
>Key: 7	Користувач натиснув клавішу 7
>Key: 8	Користувач натиснув клавішу 8
>Key: 9	Користувач натиснув клавішу 9
>Key: *	Користувач натиснув клавішу *
>Key: #	Користувач натиснув клавішу #

Описані в даному розділі команди були застосовані при програмування контролера системи та написання клієнтського ПЗ.

3.2 Розроблення програми контролера системи

Програма контролера розроблялася мовою С серед Arduino IDE версії 1.8.10 (рис. 3.1) [8].

Налагодження програми проводилося за допомогою вбудованого монітора COM порта (рис. 3.2).

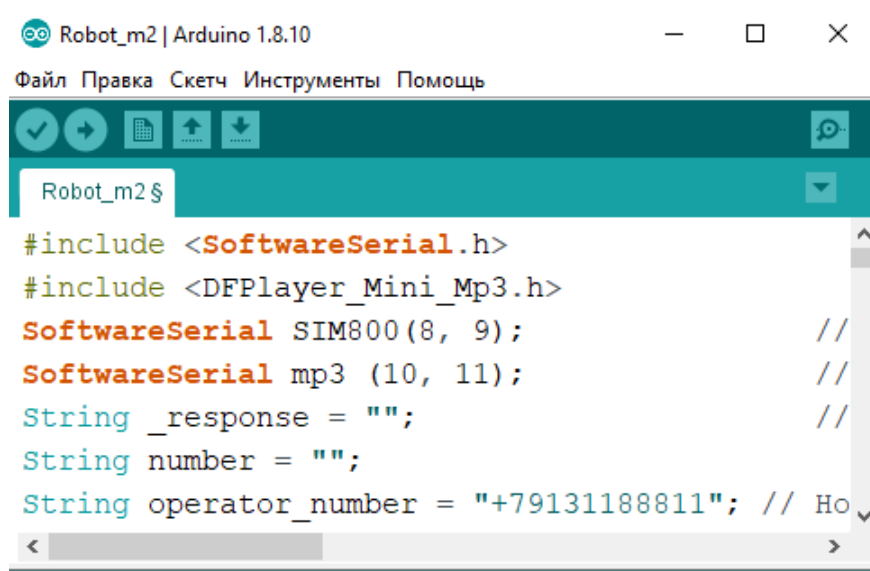


Рисунок 3.1 — Середовище Arduino IDE

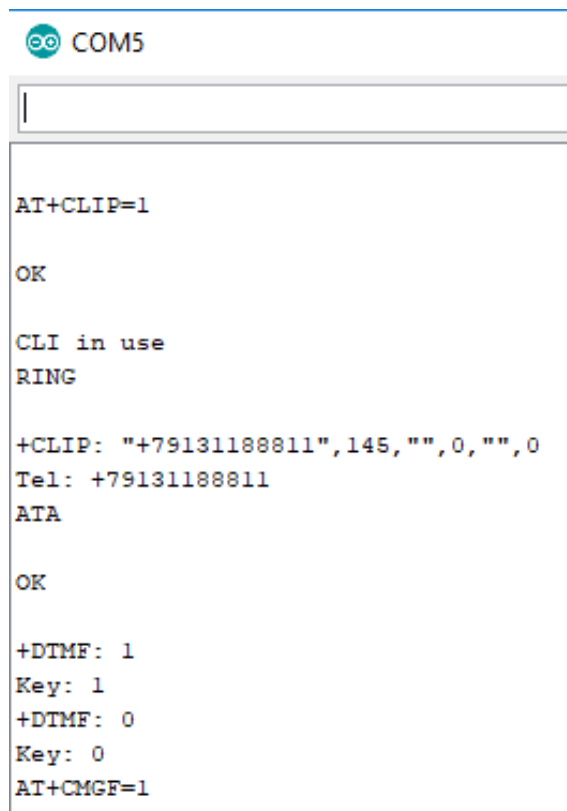


Рисунок 3.2 — Налagodження програми

Для зручного керування програвачем через послідовний порт використовується бібліотека DFPlayer_Mini_Mp3.h. У цій бібліотеці реалізовані функції управління без необхідності застосовувати шістнадцяткові команди управління. Усі необхідні функції управління представлені у табл. 3.2.

Таблиця 3.2 — Основні функції бібліотеки DFPlayer_Mini_Mp3.h

Функція	Опис
<code>mp3_set_serial();</code>	Задати порт для роботи з плеєром
<code>mp3_play();</code>	Увімкнення відтворення файлів. Як аргумент необхідно вставити номер потрібного треку
<code>mp3_next();</code>	Відтворити наступний файл
<code>mp3_prev();</code>	Відтворити попередній файл
<code>mp3_set_volume();</code>	Налаштування гучності (від 0 до 30 умовних значень)
<code>mp3_set_EQ();</code>	Налаштування еквалайзера програвача (0-5)
<code>mp3_pause();</code>	Призупинити відтворення (Пауза)
<code>mp3_stop();</code>	Зупинити відтворення (Стоп)
<code>mp3_reset();</code>	Скидання налаштувань плеєра на значення за замовчуванням
<code>mp3_single_loop (bool);</code>	Увімкнення безперервного відтворення
<code>mp3_random_play();</code>	Відтворення випадкового файлу

Відповідно до специфіки бібліотеки до коду керуючої програми контролера додано рядки, що відповідають за попереднє налаштування плеєра. Для керування GSM-модемом не потрібне підключення додаткових бібліотек. Управління всіма функціями модему здійснюється відправленням AT команд на послідовний порт модуля. Опис AT команд необхідних реалізації цього проекту умовно поділені на такі групи:

- команди налаштування модуля;
- команди для здійснення дзвінків та відправлення SMS;
- невідомі повідомлення від GSM-модему;
- команди, що передаються клієнтському ПЗ.

Сукупність застосовуваних у роботі системи команд утворюють протокол взаємодії пристрою з ПК. Детальний опис команд знаходиться у додатку М цього документа.

Функції програми контролера та його опис представлені у табл. 3.3. Повний лістинг програми представлений у додатку Х.

Таблиця 3.3 — Опис функцій програми мікроконтролера

Функція/рядки коду	Опис/призначення
voidsetup()	Функція налаштування. Функції та команди записані в цьому розділі виконуються одноразово при включення живлення контролера
SoftwareSerial SIM900(8, 9); SoftwareSerial mp3 (10, 11);	// 8 – RX Arduino (TX SIM900), 9 – TX Arduino (RX SIM900) // 10 - RX Arduino (TX MP3), 11 - TX Arduino (RX MP3) Для керування периферійними пристроями (модем, плеєр) необхідна емуляція послідовного порту двох парах цифрових висновків мікроконтролера. Відповідно до електричної схеми, представленої в додатку N, необхідно ініціалізувати підключення на 8 і 9 цифровому виведенні контролера для модему і на 10 і 11 виводі для плеєра. Апаратний SerialPort інтегрований у контролер використовується для підключення пристрої до комп'ютера

Функція / рядки коду	Опис / призначення
<pre>sendATCommand(String cmd, boolean waiting)</pre>	<p>Функція для коректного надсилання команд GSM модему</p>
<pre>waitResponse()</pre>	<p>Функція очікування відповіді модему і повернення отриманого результату. При надсиланні команд модему функція чекає на відповідь і у разі закінчення часу очікування (тайм аут) функція інформує про це користувача.</p>
<pre>SIM900.begin(9600); sendATCommand("AT", true); sendATCommand("AT+DDECT=1", true); sendATCommand("AT+CLIP=1", true);</pre>	<p>Підключення та попереднє налаштування модему. Ініціалізація порту модему. Автоматичне налаштування швидкості порту модему. Увімкнення DTMF на модемі. Включення АВН.</p>
<pre>voidloop()</pre>	<p>Основна програма контролера, яка виконується постійно, поки контролер підключений до живлення. Програма одночасно прослуховує порт модему та порт підключений до комп'ютера. Під час надходження команд з боку комп'ютера вони автоматично передаються модему. Якщо команда отримана від модему, вона дублюється в порт комп'ютера, а контролер виконує дії, запрограмовані на дану команду.</p>
<pre>if (_response.startsWith("RING")) number = _response.substring(16, 28);</pre>	<p>При вхідному дзвінку для ідентифікації необхідно отримати номер телефону клієнта</p>

Функція/рядки коду	Опис / призначення
<pre>if (_response.startsWith(h("+DTMF:")) String symbol = _response.substring(7, 8);</pre>	Якщо абонент під час дзвінка передає зворотний зв'язок, натискаючи на кнопки телефону, команда, що передається модемом, починається з "+DTMF:" Ми знаємо, що після цієї команди буде пробіл і один символ (натиснута клавіша). Записуємо отриманий символ змінної символ.
<pre>mp3_play(N);</pre>	Функція включає плеєр відтворення файлу під номером N. Файл повинен розміщуватись у папці MP3 на карті SD плеєр.
<pre>mp3_stop();</pre>	Зупинка відтворення
<pre>delay (M);</pre>	Затримка виконання програми контролера на M секунд

3.3 Програмування функцій клієнтського ПЗ

3.3.1 Загальні відомості

При реалізації системи виникла потреба написання кількох специфічних програмних функцій. Ця необхідність зумовлена вимогами, які пред'являються виробниками устаткування розробки пристроїв з урахуванням чипів даного виробника [9]. Логіка роботи пристрою представлена на рис. 3.3, 3.4.

3.3.2 Перевірка формату номера телефону на коректність

Для коректної відправки SMS повідомлення, необхідно переконатися, що введений номер телефону відповідає міжнародному стандарту. Оскільки передбачається відправка лише українським абонентам, формат номера має бути "+380xxxxxxx".

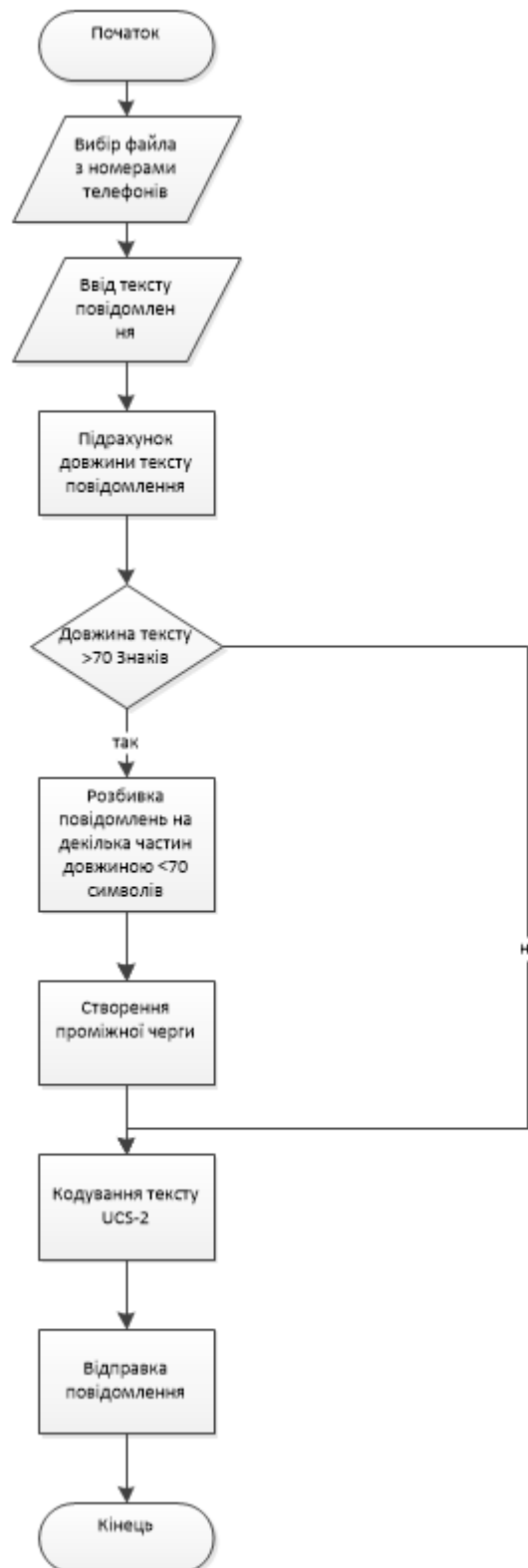


Рисунок 3.3 — Логіка надсилання повідомлень

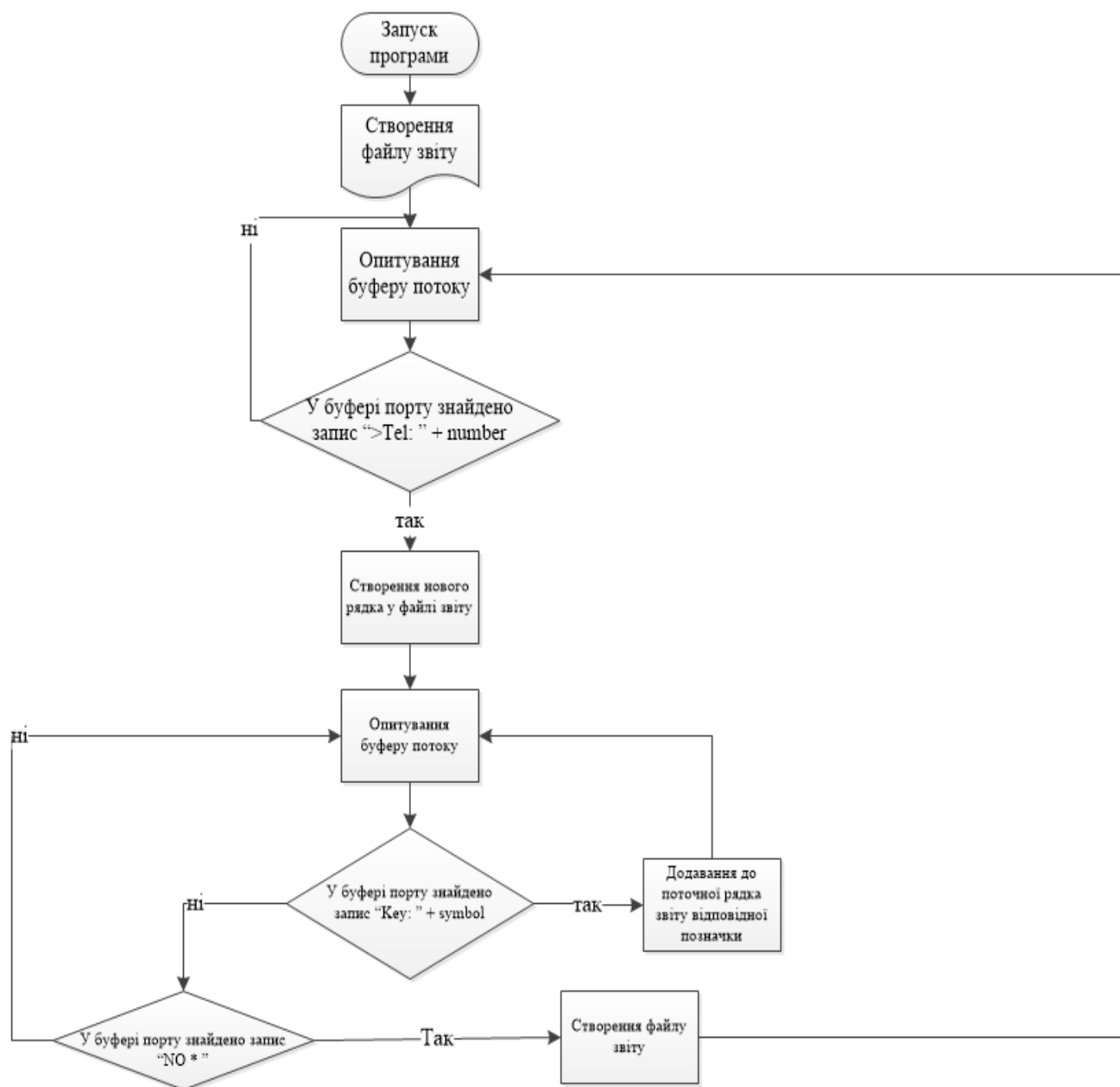


Рисунок 3.4 — Логіка обробки зворотний зв'язок

3.3.3 Надсилання україномовних SMS повідомлень

Оскільки однією з вимог до системи є можливість надсилати SMS-повідомлення українською мовою, виникає необхідність застосування протоколу PDU. Аналогічна необхідність виникне при надсиланні SMS будь-

якою мовою відмінною від англійської.

Реалізація протоколу PDU

Щоб увімкнути режим відправки SMS за протоколом PDU, необхідно передати контролеру системи команду AT+CMGF=0. Ця команда включає режим PDU для вихідних повідомлень.

Щоб надіслати повідомлення, номер телефону повинен містити лише цифри. Якщо число цифр непарне, потрібно додати в кінець номера F, сусідні цифри номера повинні бути переставлені місцями. Наприклад, номер телефону +3801188811 повинен бути перетворений так: 380118818F1.

Для реалізації цього перетворення застосовується функція, представлена на рис. 3.5.

```
// Перекодування телефонного номера для формату
PDUpublicstaticstringEncodePhoneNumber(stringPhoneNumber)
{
    stringresult ="";
    if((PhoneNumber.Length % 2) > 0) PhoneNumber += "F";

    inti = 0;
    while(i<PhoneNumber.Length)
    {
        result += PhoneNumber[i + 1].ToString() + PhoneNumber[i].ToString(); i
        += 2;
    }
    returnresult.Trim();
}
```

Рисунок 3.5 — Функція перетворення номера телефону для протоколу PDU

3.3.4 Застосування кодування UCS2

Для кодування повідомлень протоколу PDU виникає необхідність використовувати кодування USC-2. Таблиця кодів представлена рис. 3.6.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000																
0010																
0020	SP 0020	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0030	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
0060	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0070	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
0080	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0090	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
00A0	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
00B0	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	
00C0																
00D0																
0400	NBSP 00A0	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	ШY 00AD	Ў	Џ
0410	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	040A	040B	040C	040D	040E	040F
0420	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
0430	0410	0411	0412	0413	0414	0415	0416	0417	0418	0419	041A	041B	041C	041D	041E	041F
0440	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
0450	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	042A	042B	042C	042D	042E	042F
0460	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
0470	0430	0431	0432	0433	0434	0435	0436	0437	0438	0439	043A	043B	043C	043D	043E	043F
0480	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
0490	0440	0441	0442	0443	0444	0445	0446	0447	0448	0449	044A	044B	044C	044D	044E	044F
04A0	№	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ш	ў	џ
04B0	2116	0451	0452	0453	0454	0455	0456	0457	0458	0459	045A	045B	045C	00A7	045E	045F

Рисунок 3.6 — Таблица кодов UCS2 для символов кириллицы

Функцию перетворення тексту кодування UCS2 представлено нижче:

```
// Перекодування тексту в смс
UCS2publicstaticstringStringToUCS2(strings
tr)
{
UnicodeEncoding
=newUnicodeEncoding();byte[] ucs2 =
```

```
ue.GetBytes(str);  
inti = 0;  
while(i < ucs2.Length)  
{  
byteb = ucs2 [i + 1];  
ucs2[i + 1] = ucs2[i];  
ucs2[i] = b;  
i += 2;  
}  
returnBitConverter.ToString(ucs2).Replace("-", "");  
}
```

4 ТЕСТУВАННЯ СИСТЕМИ

4.1 Математична модель тестування системи

Для тестування пропонується математична модель системи, що розробляється.

Щоб теоретично оцінити продуктивність системи, було змодельовано систему масового обслуговування. В основу роботи покладено логіку робіт пристрою, що розробляється. Моделювання проводилося за допомогою бібліотеки SimEvents з пакету MATLAB. Математична модель представлена на рис. 4.1.

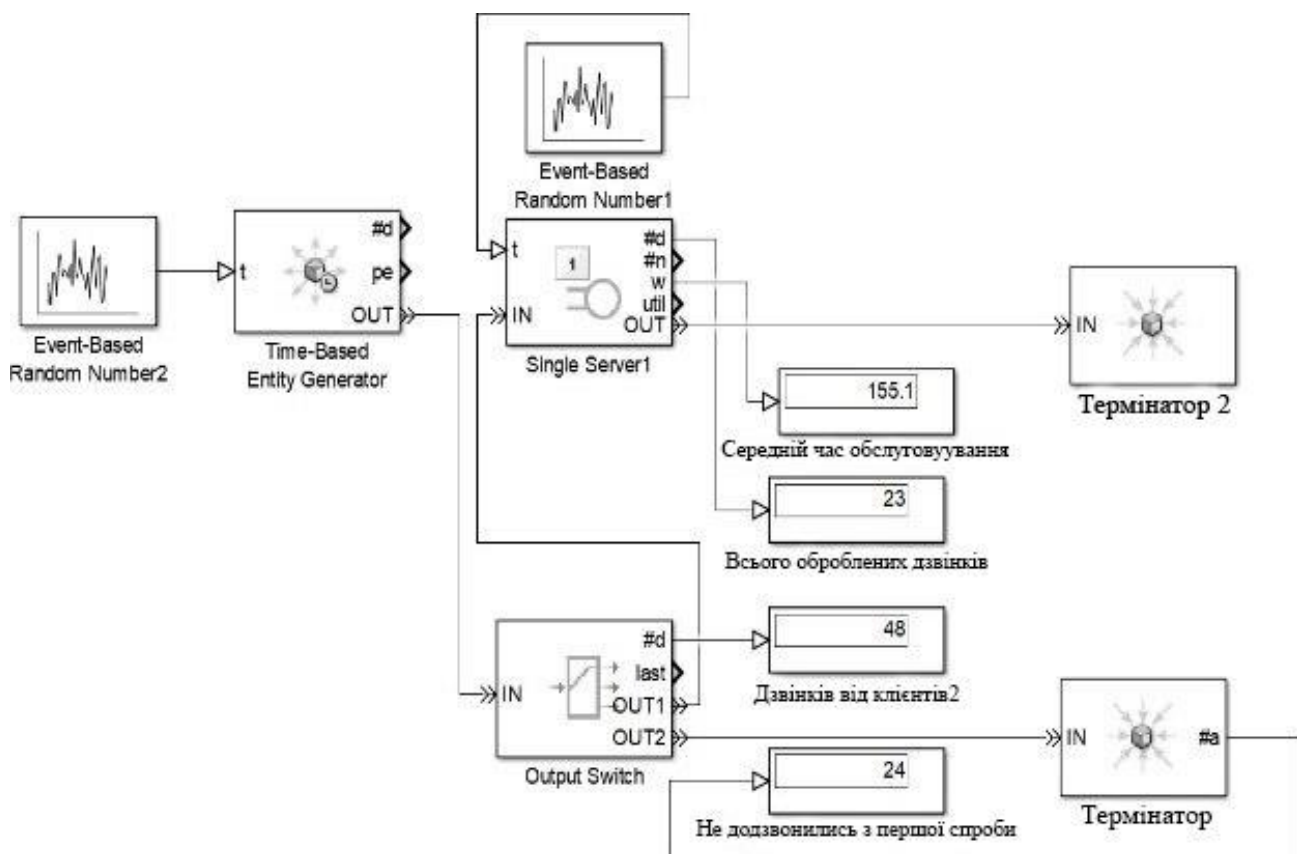


Рисунок 4.1 — Математична модель роботи одного пристрою

Для тестування застосовувались такі вихідні дані:

- час циклу тестування (симуляція) 3600 секунд;
- час обслуговування одного вхідного дзвінка 180 секунд;

— відповідно до вхідних даних, моделі були налаштовані відповідні параметри розподілів;

— в результаті тестування отримано такі дані:

— за 1 годину система обробила 23 дзвінки;

— 24 клієнти не додзвонилися з першого разу.

Отже, що застосування системи дозволить автоматично обслуговувати приблизно таку кількість клієнтів, яке секретар здатний обслужити вручну.

4.1.1 Каскадування пристроїв

На даний момент система здатна в кожний момент часу отримувати зворотний зв'язок лише від одного клієнта. Якщо на момент дзвінка клієнта 2 лінія зайнята клієнтом 1, клієнт 2 буде змушений спробувати додзвонитися пізніше. Цю ситуацію можна уникнути, якщо підключити другий аналогічний пристрій до ПК і встановити другу sim-карту. Для розподілу навантаження потрібно налаштувати переадресацію дзвінка на другу sim-карту, якщо перша sim-карта буде зайнята. Таким чином, для збільшення продуктивності можна підключати безліч пристроїв одночасно (рис. 4.2). Даний функціонал забезпечують оператори зв'язку.

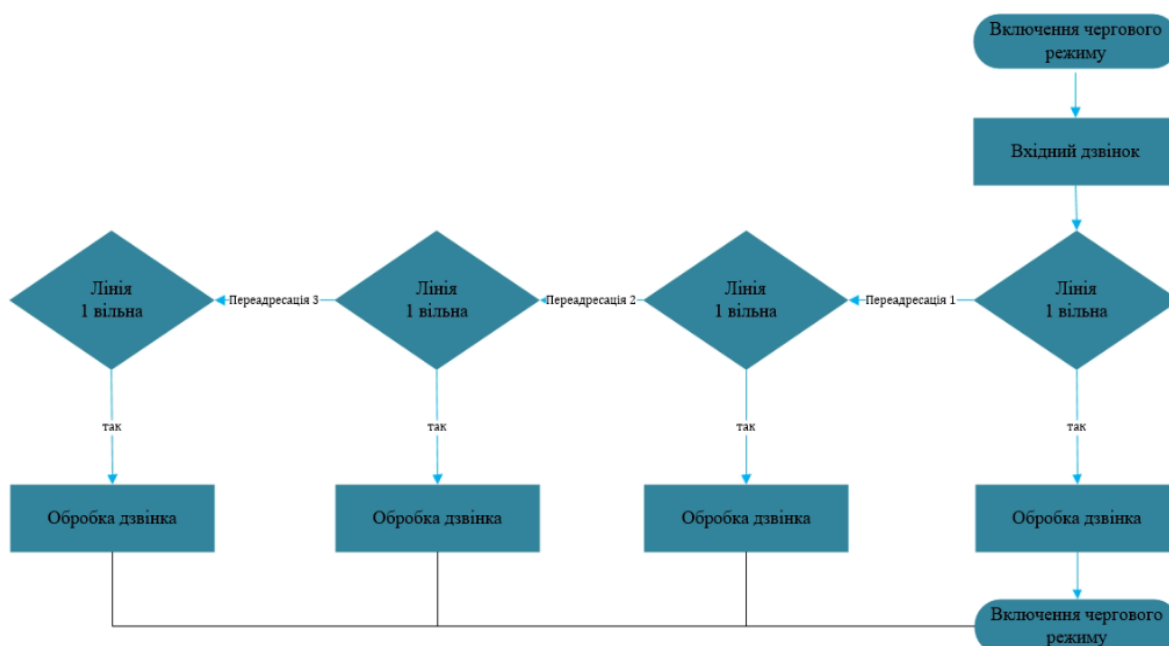


Рисунок 4.2 — Схема каскадування

При каскадуванні пристроїв показники продуктивності системи збільшуватимуться.

4.1.2 Тестування моделі каскаду із двох пристроїв

Модель представлена на рис. 4.3.

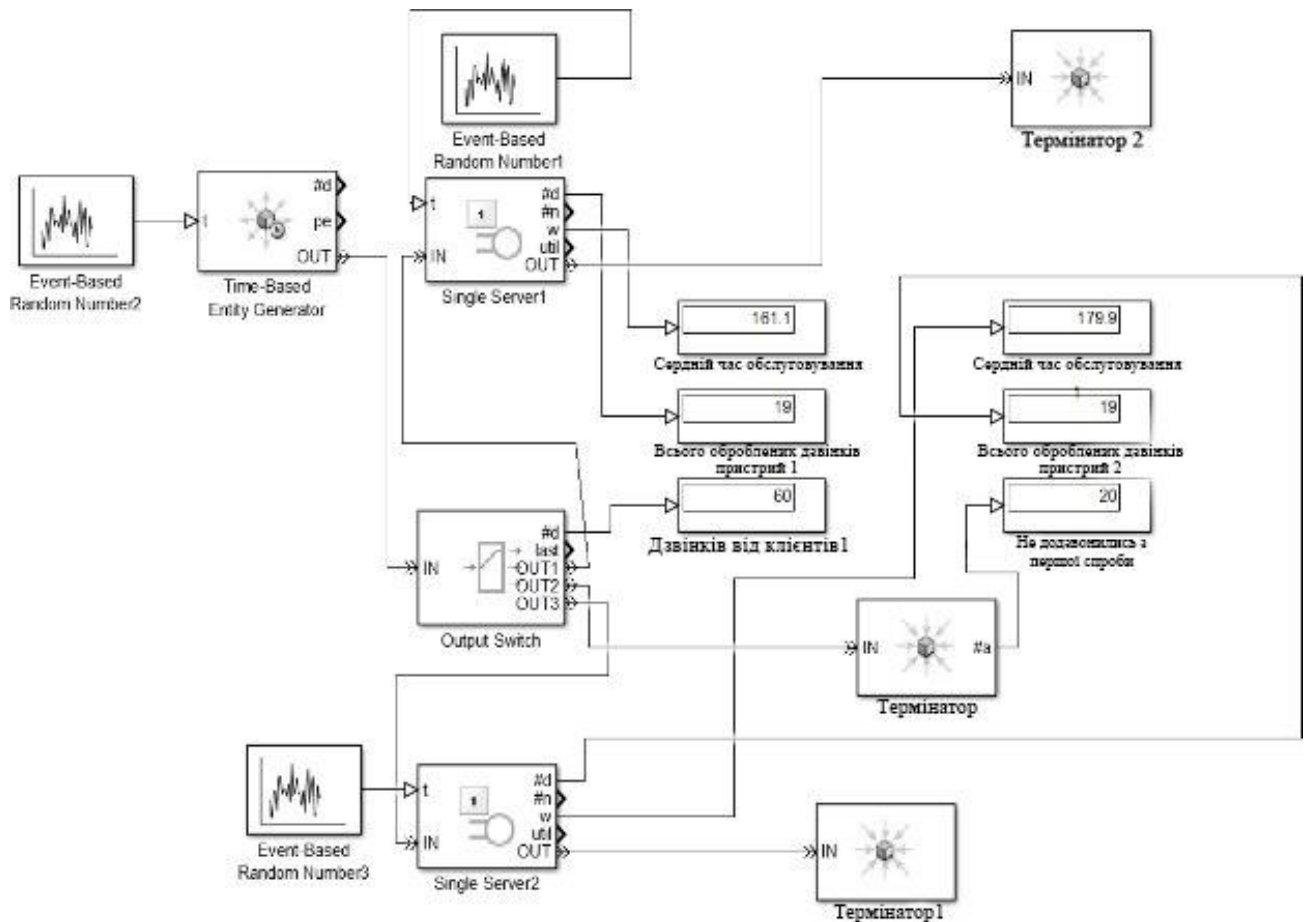


Рисунок 4.3 — Математична модель каскаду з двох пристроїв

Для тестування застосовувались такі вхідні дані:

- час циклу тестування (симуляція) 3600 секунд;
- кількість пристроїв у каскаді 2;
- час обслуговування одного вхідного дзвінка одним пристроєм із каскаду 180 секунд.

відповідно до вхідних даних моделі були налаштовані відповідні параметри

розподілів.

У результаті тестування отримано такі дані:

- за 1 годину система опрацювала 38 клієнтів;
- 20 клієнтів не додзвонилися з першого разу.

З даних випливає висновок, що застосування системи дозволить автоматично обслуговувати до 38 клієнтів на годину.

4.1.3 Результати тестування

При тестуванні моделі одиночного пристрою кількість обслужених клієнтів виявилася порівнянною з кількістю клієнтів, яку секретар здатний обслужити вручну.

При тестуванні математичної моделі каскаду з двох пристроїв система дозволить обслуговувати до 38 клієнтів на годину, що неможливо без залучення двох секретарів.

Під час проведення ручного тестування, збоїв у роботі не виявлено.

4.2 Інструкція користувача

4.2.1 Загальні відомості

В інструкції користувача описується робота програмно-апаратного комплексу контролю та інформування учасників колективної роботи. Можливі варіанти використання систем представлено на рис. 4.4.

4.2.2 Підключення апаратної частини

Перед початком роботи потрібно підключити апаратну частину системи. Вставте блок живлення пристрою в розетку та підключіть USB-кабель у вільний USB-порт ПК. ОС визначить пристрій як USB-UART-конвертер. Після

автоматичного встановлення драйвера в системі з'явиться новий COMпорт.

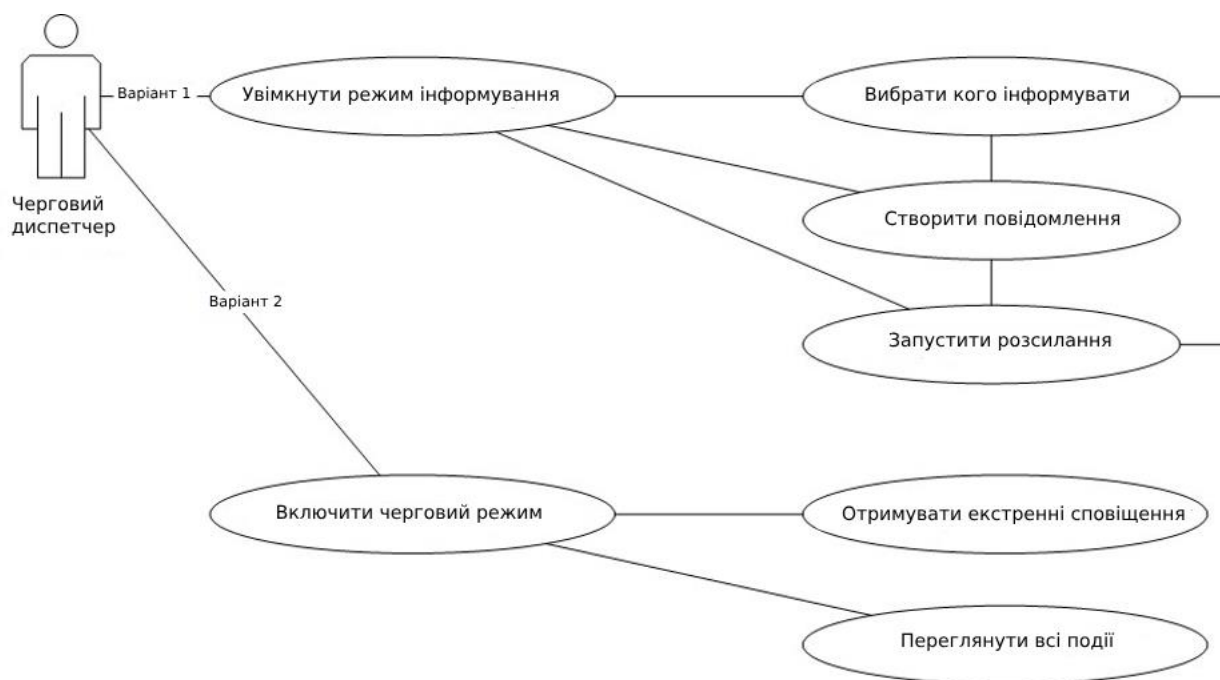


Рисунок 4.4 — Варіанти використання системи

Переглянути список встановлених у системі портів можна у розділі Порти (COM і LPT) диспетчера пристроїв (рис. 4.5).



Рисунок 4.5 — Порти (COM і LPT) у диспетчері пристроїв

Для зв'язку з пристроєм виробником контролера рекомендовано параметри середовища передачі даних, що описано в табл. 4.1

Таблиця 4.1 — Рекомендовані налаштування порту

Параметри середовища передачі даних
Швидкість обміну: 9600 бод.
Формат кадру:
1 старт біт,
8 біт даних,
1 біт парності(odd),
1 стоп біт.

Перевірити налаштування можна, натиснувши мишкою по значку Порти (COM та LPT) і перейшовши у вкладку параметри порту (рис. 4.6).

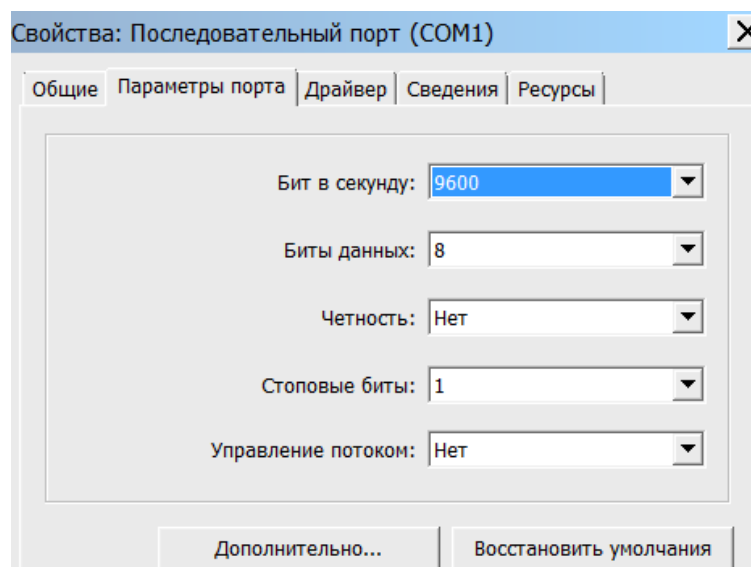


Рисунок 4.6 — Параметри порту Windows

4.2.3 Режим розсилки

У режимі розсилання система здійснює надсилання SMS повідомлень

Список одержувачів та тексти повідомлень система читає з таблиці. Від оператора системи потрібно заповнити таблицю та запустити розсилку. Для подальшої роботи системи участь оператора не потрібна. Після надсилання повідомлень програма завершить роботу.

4.2.4 Підготовка списку розсилки

Управління списком розсилки здійснюється за допомогою редагування таблиці data.xlsx. Таблиця розташовується у папці з програмою (рис. 4.7).

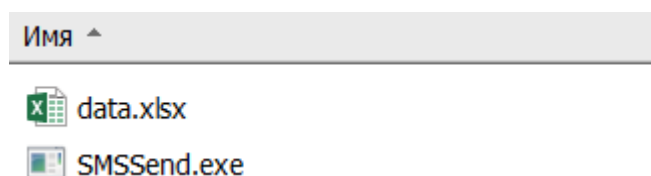


Рисунок 4.7 — Розташування таблиці data.xlsx

Для організації розсилки необхідно заповнити перший стовпець номерами телефонів, а другий стовпець внести тексти повідомлень кожного номера (рис. 4.8).

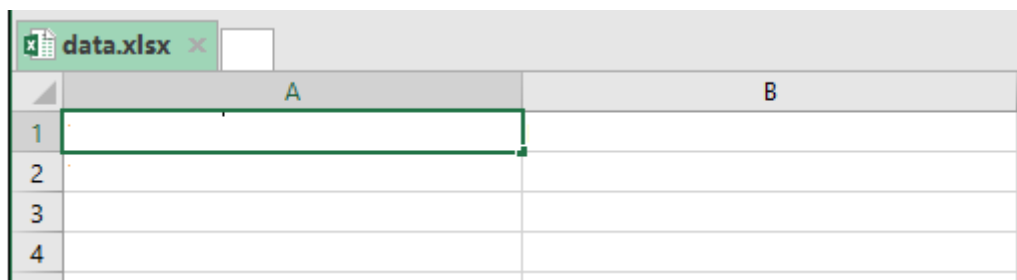


Рисунок 4.8 — Редагування таблиці data.xlsx

4.2.5 Запуск розсилки

Після підготовки файлу необхідно запустити програму SMSSend.exe. Програма пройде всіма рядками файлу data.xlsx, закодує і відправить SMS повідомлення. Роботу програми можна спостерігати у вікні терміналу – рис. 4.9.

```

CMC
AT+CMGS=39
Recived: AT+CMGS=39
>
0001000B919731118818F100081A041F044004380432043504420020042E04400438043900210021
>
+CMGS: 100
OK
Телефон -
Сообщения -
-----
AT+CMGS=47
Recived: AT+CMGS=47
>
0001000B919731118818F1000822042204350441044200200444043E0440043C04300442043000200050004400550021
>
+CMGS: 101
OK
Телефон -
Сообщения -
-----
Нажмите клавишу для выхода

```

Рисунок 4.9 — Робота програми SMSSend.exe.

У разі проблем із обладнанням програма покаже відповідне повідомлення про те, що порт закритий і його не вдалося відкрити.

У разі виникнення помилки необхідно перевірити живлення обладнання, підключення USB кабелю до порту комп'ютера та налаштування COM порту.

4.3 Режим отримання зворотного зв'язку (черговий режим)

У черговому режимі програм чекає надходження вхідних дзвінків та фіксує передані користувачем команди в excel-таблиці.

4.3.1 Увімкнення чергового режиму

Відкрийте паку з програмою та запустіть SerialPortListener.exe. З'явиться вікно терміналу, де можна спостерігати роботу програми

Після старту, програма починає отримувати інформацію дзвінках, що надійшли (рис. 4.10).

```

Listening...
Server data Received:
RING
+CLIP: "          ",14Server data Received:
Server data Received:
ATA
Server data Received:
OK
Server data Received:
+DTMF: 2
Key: 2
Server data Received:
Feedback:
Server data Received:
+DTMF: 3
Key: 3
Server data Received:
New lead:
Server data Received:
+DTMF: 1
Key: 1
Server data Received:
+DTMF: 6
Key: 6
Server data Received:
+DTMF: 0
Key: 0
Server data Received:
AT+CMGF=1
Server data Received:

```

Рисунок 4.10 — Отримання команд

Дані про всі дзвінки та натиснуті під час сеансу зв'язку клавіші заносяться до файлу звіту. Якщо інформації отриманої від робота буде недостатньо і клієнтові, що телефонує, терміново знадобиться безпосередньо поговорити з представником підприємства, контролер відправить SMS повідомлення з інформацією про дзвінок черговому співробітнику. Для цього користувач на своєму телефоні має натиснути клавішу 0 (рис. 4.11).

```

Key: 0
Server data Received:
AT+CMGF=1
Server data Received:
OK
AT+CMGS="          "
Server data Received:
>
Server data Received:
TEST

```

Рисунок 4.11 — Запит дзвінка від користувача

Поточний файл звіту можна переглянути після завершення роботи програми.

4.3.2 Аналіз зворотного зв'язку

Для перегляду звітів програми відкрийте папку з програмою і оберіть звіт, який Вас цікавить. Для іменування файлу звіту використовується формат log_MM_DD_YYHHMM. Файли звітів створюються під час кожного запуску програми (рис. 4.12).

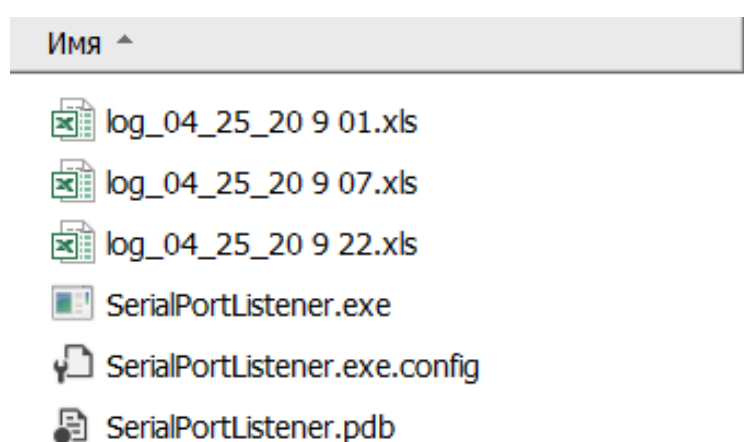


Рисунок 4.12 — Файли звітів

Файл звіту складається зі списку номерів телефонів, від яких були прийняті дзвінки та отримані від них команди (рис. 4.13).

log_05_02_20 17 22.xls								
	A	B	C	D	E	F	G	H
1	Номер телефона	Команда 1	Команда 2	Команда 3	Команда 4	Команда 5	Команда 6	Команда 7
2		2	3	1	6	0		

Рисунок 4.13 — Зміст файлу звіту

По сукупності команд можна зробити висновок про наміри людини телефонувати і прийняти рішення про подальші дії.

4.4 Системні вимоги

Таблиця 4.2 – Системні вимоги

Категорія	Вимоги
Процесор	32- або 64-розрядний процесор з тактовою частотою 2 ГГц
Операційна система	Windows 7 або пізніша версія
Оперативна пам'ять	1024 МБ
Вільне місце на жорсткому диску	100 МБ вільного місця на диску
Монітор	Стандартний VGA монітор з роздільною здатністю 800 x 600 або вище
Клавіатура	Стандартна клавіатура
Порти	Наявність порту USB
Програмне забезпечення	-MicrosoftOffice 2017 або пізніші версії -Microsoft .NET Framework 7.5 або пізніші версії

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку апаратно-програмної системи контролю та інформування учасників колективної роботи.

Особливістю апаратно-програмної системи є те, що дана технологія використовує удосконалений метод контролю, а також інноваційні засоби його реалізації на основі сучасних апаратних і програмних засобів, що його реалізують, що підвищує швидкість інформування учасників колективної роботи.

Аналогом даної апаратно-програмної системи контролю та інформування може бути Hidrolock GSM за ціною 5640 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження табл.5.1

Критерій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

Критерій	0	1	2	3	4
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблицю 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ШБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	2	3	3
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	3	3	3
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	3	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	2	3	3
Сума	39	40	41
Середньоарифметична сума балів	$(39+40+41) / 3 = 40$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 -20	Нижче середнього
21 -30	Середній
31 -40	Вище середнього
41 -48	Високий

Як видно з табл. 5.3, рівень комерційного потенціалу розроблюваного апаратно-програмної системи контролю та інформування учасників колективної роботи є високим, що досягається за рахунок того, що дана система відрізняється від існуючих тим, що дана технологія використовує удосконалений метод контролю, а також інноваційні засоби його реалізації на основі сучасних апаратних і програмних засобів, що його реалізують, що підвищує швидкість та точність інформування учасників колективної роботи.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 22 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.1.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	21500	977,27	28	27363,636
Програміст	20000	909,09	28	25454,545
Всього				52818,18

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 10 \% / 100 \% \quad (5.2)$$

$$Z_d = (52818,18 \cdot 10 \% / 100 \%) = 5281,82 \text{ (грн.)}$$

Нарахування на заробітну плату розробників. Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (52818,18 + 5281,82) \cdot 22 \% / 100 \% = 12782,00 \text{ (грн.)}$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки. Амортизація обладнання, що використовувалось для розробки в спрощеному

вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_{в}} \cdot \frac{t_{вик}}{12} \text{ [грн.]}. \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,27 міс.

$$A_{обл} = \frac{20000}{2} \times \frac{1,27}{12} = 1060,61 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до табл. 5.2.

Таблиця 5.5 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (Ноутбук Dell)	20000	2	1,27	1060,606
Офісне обладнання (меблі)	22000	4	1,27	583,333
Приміщення	750000	20	1,27	3977,273
Всього				5621,21

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних

нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив (Microsoft Windows 10) не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 1100$ грн.

Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,35$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,35 \cdot 8 \cdot 28 \cdot 6,2 = 437,472 \text{ (грн.)}$$

Інші витрати та загальновиробничі витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ів}}{100\%}, \quad (5.6)$$

де $H_{\text{ІВ}}$ – норма нарахування за статтею «Інші витрати».

$$I_{\text{В}} = 52818,18 * 50\% / 100\% = 26409,09 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нзв}} = (3_o + 3_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.7)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 52818,18 * 100\% / 100\% = 52818 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи. Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$\begin{aligned} V_{\text{заг}} &= 52818,18 + 5281,82 + 12782,00 + 5621,21 + 1100 + 437,47 + 26409,09 + \\ &+ 52818 = 157267,96 \text{ грн.} \end{aligned}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів. Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ЗВ, визначається за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 157267,96 / 0,5 = 314536 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного

інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем. У цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій річній підписці в 2500 грн. термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 100 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 5000 шт., протягом другого року – на 3000 шт., протягом третього року на 2000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*100 + (2500 + 100) * 5000) * 0,8333 * 0,25 * (1 - 0,18) = 2135416,581 \text{ грн.}$$

$$\Delta\Pi_2 = (0*100 + (2500 + 100) * (5000+3000)) * 0,8333 * 0,25 * (1 - 0,18) = 3553333,191 \text{ грн.}$$

$$\Delta\Pi_3 = (0*100 + (2500 + 100) * (5000+3000+2000)) * 0,8333 * 0,25 * (1 - 0,18) = 4441666,489 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 10130416,26 грн

Розрахунок ефективності вкладених інвестицій та періоду їх окупності. Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких

виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T — період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\text{ПП} = (2135416,581/(1+0,1)^1) + (3553333,191/(1+0,1)^2) + (4441666,489/(1+0,1)^3) = 1941287,80 + 2936639,001 + 3337089,774 = 8215016,576 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} * ЗВ, \quad (5.11)$$

де $k_{\text{інв}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 2 \dots 5$, але може бути і більшим;

$ЗВ$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 314536 = 629071,83 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV , Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = \text{ПП} - \text{PV}, \quad (5.13)$$

$$E_{abc} = 8215016,576 - 629071,83 = 7585944,75 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формулу:

$$E_g = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_g = \sqrt[3]{(1 + 7585944,75/629071,83) - 1} = 1,355$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (5.16)$$

$$T_{ок} = 1 / 1,355 = 0,74 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,74 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 314536 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблена апаратно-програмна система за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,74 роки.

ВИСНОВКИ

У роботі було розроблено апаратно-програмну систему контролю та інформування учасників колективної роботи. Для цієї мети було проведено перед проектне дослідження організації. На етапі передпроектних досліджень було проведено аналіз та реінжиніринг бізнес-процесу роботи з клієнтами. На підставі отриманої інформації було створено технічне завдання, відповідно до якого була розроблена апаратна та програмна частина системи, а також пакет прикладного ПЗ. яке показало, що система без участі користувача здатна обслуговувати 23 клієнти на годину.

Було створено система, яка має такі конструктивні, технологічні та техніко-експлуатаційні характеристики:

- швидкість інформування 300 абонентів за годину;
- швидкість обробки зворотного зв'язку від 23 клієнтів за годину;
- габарити: 100x60x20 мм;
- напруга живлення DC 4.2 - 24, AC 220 В;
- споживаний струм до 2.5 А;
- підтримка діапазонів GSM 850,900,1800,1900 МГц;
- інтерфейси UART, USB, DC, Audio, слот SD-карти, слот SIM-карти;
- підтримка протоколів TCP/UDP, SSL, HTTP/FTP, SMS, MMS, E-Mail, DTMF;
- можливість каскадування пристроїв для збільшення швидкості розсилки та обробки зворотного зв'язку.

Розроблена система дозволяє користувачеві підготувати список абонентів для розсилки, автоматично надіслати повідомлення всім вказаним у списку абонентам, та отримати зворотний зв'язок за допомогою автоматичного прийому вхідних дзвінків. Під час надходження вхідного дзвінка система автоматично відповість на вхідний дзвінок, озвучить повідомлення та передасть на ПК інформацію про дії абонента (натискання кнопок телефону) під час сеансу зв'язку.

Можливість каскадування пристроїв для збільшення швидкості розсилки та обробки зворотного зв'язку.

У результаті дослідження було виконано моделювання та реінжиніринг бізнес-процесів. Розроблено мікропроцесорний пристрій, здатний автоматизувати більшість процесів, пов'язаних з інформуванням клієнтів. Для керування пристроєм на мові C# розроблено пакет прикладних програм. Проведено тестування математичної моделі системи.

Для виконання роботи та підготовки пояснювальної записки використовувалося таке ПЗ:

- Microsoft Windows 10 Professional;
- Microsoft Word 2016;
- Microsoft Excel 2016;
- Microsoft PowerPoint 2016;
- Microsoft Visio Professional 2016;
- Microsoft Visual Studio 2017;
- ArduinoIDE 1.8.12;
- KiCad 5.1.4.
- TheMathWorksMATLABR2015b

У процесі розроблення застосовано мову програмування C#.

Були доведено економічну доцільності розробки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сало І. В. Контроль у системі управління. Видавництво: Університетська книга. ISBN 978-966-2704-03-7.
2. Гріфін Р., Яцура В. Основи менеджменту: Підручник / Наук.ред. В.Яцура, Д. Олесневич. Львів: БаК, 2001. 624 с.
3. Назарчук Т. В., Косіюк О. М. Менеджмент організацій: навч. посіб. К.: «Центр учбової літератури», 2015. 560 с.
4. AccompanytoSIMtech. SIM900HL_Hardware_Design_V2.01C. URL: https://www.elecrow.com/download/SIM900C_Hardware_Design_V1.02.pdf (Дата звернення 05.06.2023)
5. DFPLayer. Serial MP3 module. Module Application Instruction. URL : <http://ardu.net/uk/moduli/264-zvukovoj-modul-dfplayer-mini-mp3-module-arduino-mp3-tf-16p-yx5200.html> (Дата звернення 05.06.2023).
6. ArduinoNano-Rev3.2.sch. URL: https://www.arduino.cc/en/uploads/Main/Arduino_Nano-Rev3.2-SCH.pdf (Дата звернення 05.06.2023)
7. MT6261DGSMGPRSSOCProcessorTechnicalBrief. URL: <https://datasheetspdf.com/pdf-file/1266527/MEDIATEK/MT6261D/1> (Дата звернення 05.06.2023)
8. Simon Monk. Programming Arduino. Getting Started with Sketches. URL: <http://index-of.es/Varios-2/Programming%20Arduino.pdf> (Дата звернення 05.06.2023).
9. A company to SIM tech. SIM900 Series_AT Command Manual_V1.09. URL:https://www.elecrow.com/wiki/images/2/20/SIM900_Series_AT_Command_Manual_V1.09.pdf (Дата звернення 05.06.2023).
10. Texas Instruments. MAX232x Dual EIA-232 Drivers/Receivers URL:<http://www.ti.com/lit/gpn/max232> (Дата звернення 05.06.2023).
11. KiCad. Посібник для освоєння основних принципів успішної розробки друкованих плат електронних пристроїв у системі автоматизованого проектування KiCad. URL: https://docs.kicadpcb.org/4.0.7/ua/getting_started

[_in_kicad/getting_started_in_kicad.html](#) (Дата звернення 05.06.2023).

12. TheDataSheetofCH340. URL: <https://www.mpja.com/download/35227/cpdata.pdf> (Дата звернення 05.06.2023).

13. General questions, відношення до UTF або Encoding Form. URL:
1.http://www.unicode.org/faq/utf_bom.html#utf16-11 (Дата звернення 05.06.2023).

Додаток А
Технічне завдання
Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ
Завідувач кафедри ОТ
_____ проф., д.т.н. О. Д. Азаров
« ___ » _____ 20__ р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
«Апаратно-програмна система контролю та інформування учасників колективної
роботи»
08-23.МКР.002.00.000 ПЗ

Науковий керівник
к.т.н., проф. каф. ОТ
_____ Азарова А. О.
виконав:
магістрант 2 курсу,
_____ Годун. В. В.

1. Підстава виконання магістерської кваліфікаційної роботи

1.1 Особливу увагу при оптимізації їх функціонування організації звертають на інформаційні технології. Така увага зумовлена можливістю перекласти виконання різного роду завдань з людини на автоматизовану систему, обслуговування якої обходиться організації дешевше, ніж утримання співробітника, який виконує покладені на нього обов'язки. Сучасні технології дозволяють це робити, а тому створення такого роду систем є актуальним.

1.2 Наказ про затвердження теми магістерської кваліфікаційної роботи від 20.03.2023 р. № 68.

2 Мета і призначенням МКР

2.1 Метою роботи є аналіз створення апаратно-програмної системи контролю та інформування учасників колективної роботи.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3. Вихідні дані для виконання МКР

Вихідні дані для виконання МКР: процесор Core i7 5930K, Відеокарта не гірша за Radeon RX 480 G1, ОЗП не менше 32 ГБ із тактовою частотою 3200 МГц і вище, SSD накопичувач.

4. Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- забезпечити протокол взаємодії апаратної частини з комп'ютером;
- провести моделювання та тестування системи;
- створити керівництво користувача.

— 5. Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Табл. А.1.

Таблиця А.1 — Етапи МКР

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Аналітичний огляд предметної області	21.03.21	
2	Аналіз способів контролю та інформування	22.03-02.04.23	
3	Аналіз вимог	10.04-08.04.23	
4	Розробка апаратної частини системи	09.04-29.04.23	
5	Розробка ПЗ	30.04-19.05.23	
6	Тестування системи	20.05-25.05.23	
7	Розрахунок економічної частини роботи	26.05-.31.23	
8	Оформлення пояснювальної записки та ілюстративного матеріалу	01.06-04.06.2.23	
9	Аналіз виконання роботи, висновки, додатки	05.06-06.06.23	
10	Перевірка якості виконання магістерської роботи та усунення недоліків	08.06.23	

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022.

Додаток Б

Електрична схема апаратної частини системи

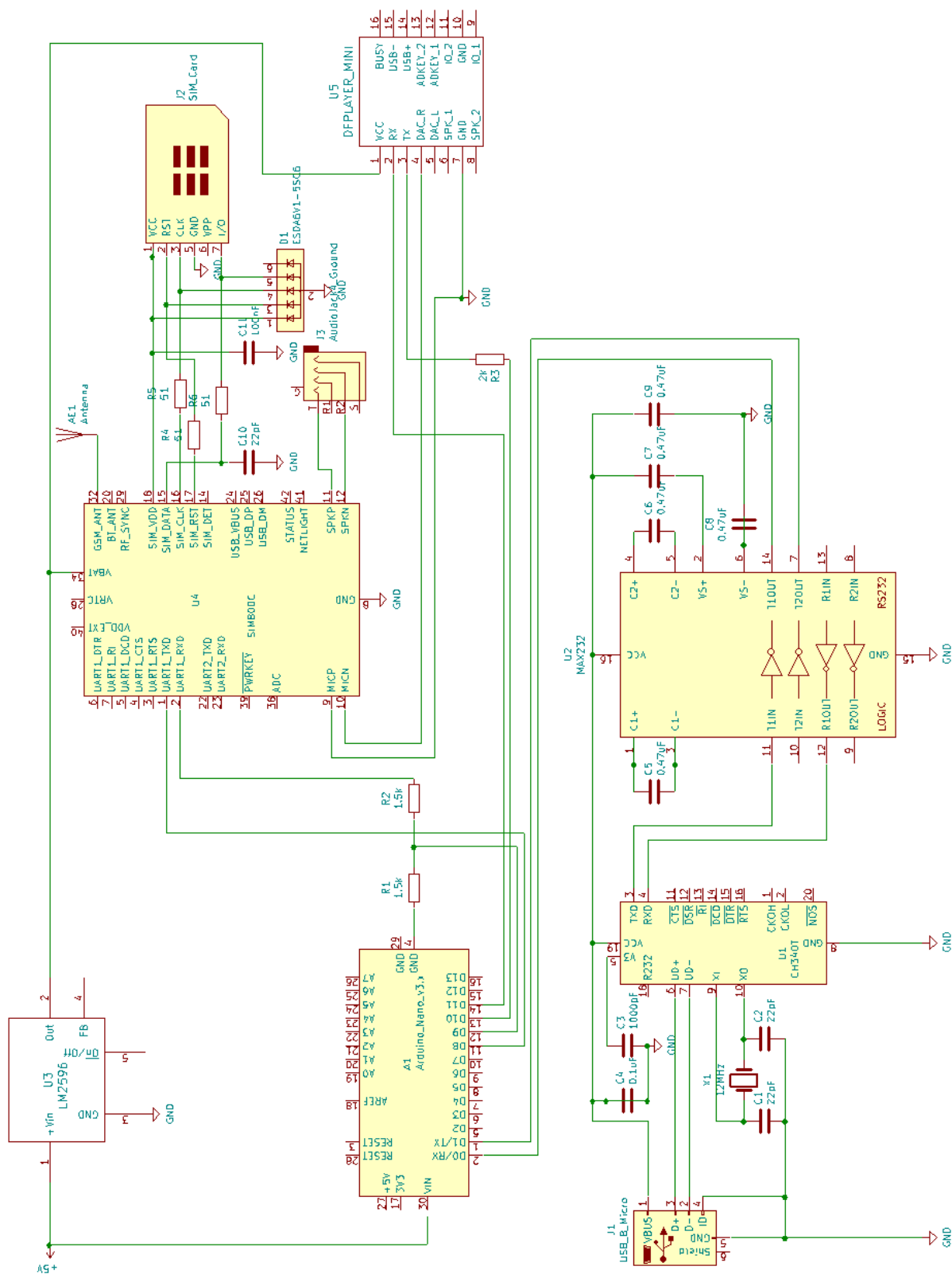


Рисунок Б1 — Електрична схема апаратної частини системи

Додаток В
Список електронних компонентів

Позначення на схемі	Прилад	Кількість
A1	Arduino_Nano_v3.x	1
AE1	Antenna	1
C3	1000pF	1
C4	0.1uF	1
> C5-C9	0.47uF	5
> C1, C2, C10	22pF	3
C11	100nF	1
D1	ESDA6V1-5SC6	1
J1	USB_B_Micro	1
J2	SIM_Card	1
J3	AudioJack4_Ground	1
> R1, R2	1.5k	2
R3	2k	1
> R4-R6	50	3
U1	CH340T	1
U2	MAX232	1
U3	LM2596	1
U4	SIM900C	1
U5	DFPLAYER_MINI	1
X1	12MHz	1

Рисунок В1 — Список електронних компонентів

Додаток Г

Команди налаштування GSM-модуля

Опис	Команда	Структура відповіді	Примітки приклади
Опитування стану модуля	AT	AT OK	OK
Інформація про пристрій	ATI	<info> OK <info> — модель та версія модуля	SIM900 R14.18 OK
Готовність модуля здійснювати дзвінки	AT+CCAL R?	+CCALR: <mode> OK <mode> - ідентифікатор готовності: 0 — модуль не готовий здійснювати дзвінки 1 — модуль готовий здійснювати дзвінки	+CCALR: 1 OK
Запит якості зв'язку	AT+CSQ	+CSQ: <rssi>,<ber> OK <rssi> - якість сигналу (від 10 і вище - нормальна): 0 -115 дБм і менше 1 -111 дБм 2...30 -110... -54 дБм	+CSQ: 8,0 OK

Опис	Команда	Структура відповіді	Примітки приклади
		31 -52 дБм та більше 99 визначити неможливо <ber> - RXQUAL (захід якості сигналу)	
Запросити ІМЕІ	AT+GSN	<sn> ОК <sn> - ІМЕІ модуля	8395230754308 92 ОК
Запит напруги живлення	AT+CVC	+CVC: <bc>,<bcl>,<voltage> ОК <bc> - статус зарядки 0 - зарядки немає 1 - зарядка йде 2 - зарядка завершена <bcl> — обсяг заряду, що залишився у відсотках (1...100) <voltage> - напруга живлення модуля, в мілівольтах	+CVC: 0,73,3988 ОК
Тип реєстрації у мережі	AT+CREG?	+CREG: <n>,<stat> ОК <n> — параметр відповіді 0 — код, що не запитується реєстрації в мережі вимкнено 1 — незапитуваний код реєстрації в мережі увімкнено 2 — код, що не запитується	+CREG: 0,1 ОК

Опис	Команда	Структура відповіді	Примітки приклади
		<p>реєстрації в мережі увімкнено з інформацією про місцезнаходження</p> <p><stat> - статус</p> <p>0 - незареєстрований, не шукає нового оператора для реєстрації</p> <p>1 — зареєстрований у домашній мережі</p> <p>2 — незареєстрований, але у пошуку нового оператора для реєстрації</p> <p>3 — реєстрацію заборонено</p> <p>4 — невідомо</p> <p>5 - зареєстрований, у роумінгу</p>	
Інформація про оператора	AT+COPS?	<p>+COPS: <mode>,[<format>,<operator>] OK</p> <p><mode> — режим реєстрації</p> <p><format> — формат відображення</p> <p><oper> - найменування оператора у заданому форматі</p>	<p>+COPS: 0,0,"MTS"</p> <p>OK</p>
Статус телефону	AT+CPAS	<p>+CPAS: <pas></p> <p>OK</p>	+CPAS: 4

Опис	Команда	Структура відповіді	Примітки приклади
GSM-модуля		<p><pas> - статус</p> <p>0 - готовий до виконання команд з терміналу</p> <p>2 - невідомо (виконання команд не гарантується) 3 - вхідний виклик, але модуль готовий до виконання команд</p> <p>4 — вихідний дзвінок, але модуль готовий до виконання команд</p>	OK
Налаштування швидкості порту	AT+IPR	<p>AT+IPR =<n></p> <p><n> - швидкість прта 0 - авто визначення 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200</p>	
Встановлення парності	AT+ICF	<p>AT+ICF=<format>[,<parity>]</p> <p><format></p>	

Опис	Команда	Структура відповіді	Примітки приклади
порту модуля		1 8 data 0 parity 2 stop 2 8 data 1 parity 1 stop 3 8 data 0 parity 1 stop 4 7 data 0 parity 2 stop 5 7 data 1 parity 1 stop 6 7 data 0 parity 1 stop <parity> 0 odd 1 even 3 space (0)	
Дублювання в терміналі відправлено на модуль команди	ATE	ATE<value> <value> 0 - вимкнений, 1- Увімкнено(позамовчуванням)	

Додаток Д

Лістинг програми надсилання повідомлень

```
using System;
using System.Collections.Generic;
using System.IO;
using System.IO.Ports;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Text.RegularExpressions;
using Excel = Microsoft.Office.Interop.Excel;

namespace SMS_from_file
{
    static class Program
    {
        static SerialPort port = new SerialPort();

        static void Main(string[] args)
        {
            // Локація папки
            string executableLocation =
                Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location);

            // Локація xls файлу
            string xlsDataLocation = Path.Combine(executableLocation, "data.xlsx");

            // створюємо файл якщо не існує в
            папці if (!File.Exists(xlsDataLocation)) File.
```

```
Create (xlsDataLocation);

// всі телефонні номери з
докstring[] phones =null;
// всі повідомлення з
докstring[] messages
=null;
WriteData(xlsDataLocation,refphones,refmessages);
OpenPort();
if(phones! =null&& messages !=null)
{
foreach(varcolumnsinphones.Zip(messages,Tuple. Create))
{
boolresult = sendSMS(columns.Item2,
columns.Item1);if(result ==true)
{
Console.WriteLine($"Телефон -
{columns.Item1 }");Console.WriteLine($"Повідомлення -{ columns.Item2}");

Console.WriteLine(newstring('-', 20));
}
else
{
Console.Write("не отруєний");
}
}
}

Console.WriteLine("Натисніть клавішу, щоб
вийти");Console.ReadKey();
```

```
}
```

```
///

```

```
///

```

```
///

```

```
///

```

```
///

```

```
///

```

```
privatestaticboolsendSMS(stringtextsms,stringtelnumber)
```

```
{
```

```
try
```

```
{
```

```
if(!port.IsOpen)
```

```
{
```

```
Console.WriteLine("Портзакритий!");returnf
```

```
else;
```

```
}
```

```
telnumber = telnumber.Replace("-",string.Empty).Replace(" "
```

```
",string.Empty).Replace("+",string.Empty);
```

///

01 це PDU Type або іноді називається SMS-SUBMIT. 01 означає, що повідомлення, що передається, а не одержуване

///

цифри 00 це TP-Message-Reference означають, що телефон/модем може встановити кількість успішних повідомлень автоматично

///

telnumber.Length.ToString("X2") видасть нам довжину номера у 16-річному форматі

///

91 означає, що використовується міжнародний формат телефонного номерatelnumber ="01"+"00"+ telnumber.Length.ToString("X2") +"91"+ EncodePhoneNumber(telnumber);

```

varsubText = SplitInParts(textsms,
70).ToArray();Array.ForEach(subText, str =>
{
varsendSrt = StringToUCS2(str);

///00 означає, що формат повідомлення неявний. Це ідентифікатор протоколу.
Інші варіанти: телекс, телефакс, голосове повідомлення тощо.
///08 означає формат UCS2 - 2 байти на символ. Він простіше, тож
розглянемо його.
///якщо замість 08 вказати 18, повідомлення не буде збережено на телефоні.
Вийде flash повідомлення
stringlengthOfByte = (sendSrt.Length / 2).ToString("X2");
sendSrt = telnumber + "00"+"08"+ lengthOfByte + sendSrt;
///посилаємо команду з довгою повідомлення – кількість октет у десятковій
системі. тобто ділимо на дві кількості символів у повідомленні
///якщо октет неповний, то вийде в результаті дробове число. це
дробове число округляємо до більшого
stringlengthOfMessage =Math.Ceiling((double)(sendSrt.Length / 2)).ToString();
port.Write($"AT+CMGS={lengthOfMessage}\r\n");
System.Threading.Thread.Sleep(2000);Console.WriteLine($"AT+CMGS={length
OfMessage}\r\n");
///номер sms-центру ми не вказуємо, вважаючи, що практично у всіх SIM
картах він уже прописаний
///для того, щоб було зрозуміло, що цей номер ми не вказали додаємо до
нашого повідомлення на початок 2 нуля
///додаємо саме ПІСЛЯ того, як підраховали довжину
повідомленняport.Write($"00 {sendSrt}\u001a\r\n");

System.Threading.Thread.Sleep(10000);

```

```

});
#regionrecieved data

stringrecievedData;
recievedData = port.ReadExisting();

Console.WriteLine($"Recived:{recievedData}");if(
RecievedData.Contains("ERROR"))
    {
    returnfalse;
    }

#endregion
}
catch(Exceptionex)
{
Console.WriteLine("Помилкапривідпаковціповідомлення", ex.
Message);returnfalse;
}

returntrue;
}

privatetaticvoidOpenPort()
{
port.BaudRate = 9600;// ще варіанти 4800, 9600, 28800 або 56000
port.DataBits = 8;// ще варіанти 8, 9

//port.StopBits = StopBits.One; // ще варіанти StopBits.TwoStopBits.None
або StopBits.OnePointFive

```



```
//port.Parity = Parity.Odd; // ще варіанти Parity.Even Parity.Mark Parity.None
або Parity.Space
```

```
port.ReadTimeout = 2500; // ще варіанти 1000, 2500 чи 5000 (більше не варто)
```

```
port.WriteTimeout = 2500; // ще варіанти 1000, 2500 чи 5000 (більше не варто)
```

```
//port.Handshake = Handshake.RequestToSend;
```

```
//port.DtrEnable = true;
```

```
//port.RtsEnable = true;
```

```
//port.NewLine = Environment.NewLine;
```

```
//port.Encoding = Encoding.GetEncoding("");
```

```
port.Encoding = Encoding.GetEncoding("windows-1251");
```

```
port.PortName = "COM8";
```

```
// конструкція для відкриття
```

```
порту if (port.IsOpen)
```

```
port.Close(); try
```

```
{
```

```
{
```

```
port.Open();
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
Console.WriteLine("Не вдалося відкрити порт", ex.Message);
```

```
}
```

```
}
```

```
public static string EncodePhoneNumber(string PhoneNumber)
```

```
{
```

```
string result = "";
```

```
if ((PhoneNumber.Length % 2) > 0) PhoneNumber += "F";
```

```

inti = 0;
while(i < PhoneNumber.Length)
{
result += PhoneNumber[i + 1].ToString() + PhoneNumber[i].ToString(); i
+= 2;
}
returnresult.Trim();
}

```

```

///<summary>
///перекодування тексту смс у UCS2
///</summary>
///<param name="str"></param>
///<returns></returns>
publicstaticstringStringToUCS2(stringstr)
{
UnicodeEncodingue
=newUnicodeEncoding();byte[] ucs2 =
ue.GetBytes(str);

inti = 0;
while(I < ucs2.Length)
{
byteb = ucs2 [i + 1];
ucs2[i + 1] = ucs2[i];
ucs2[i] = b;
i += 2;
}
returnBitConverter.ToString(ucs2).Replace("-", "");

```

```

}

///

```

```

Формат: 5,
Password:string.Empty,
WriteResPassword:string.Empty,
IgnoreReadOnlyRecommended:true, Origin:
Excel.XIPlatform.xlWindows, Delimiter:"\t",
Editable:false,
Notify:false,
Converter: 0,
AddToMru:true);
// створення листа
varsheets = wb.Worksheets;

// створити робочий лист
varws = (Excel.Worksheet)sheets[1];

// перший стовпець (телномірів)
varfirstCalls = (Array)ws.UsedRange.Columns[1].Cells.Value;

// Втаройстовпець (сполучення)
varlastCalls = (Array)ws.UsedRange.Columns[2].Cells.Value;

if(firstCalls !=null&& lastCalls !=null)
{
// тел:
phones = firstCalls.OfType<object>().Where(phone =>
{
varregex =@"^((8|\+7)[\ - ]?)?(\(?\d{3}\)?[\ - ]?)?[\d\ -
]{7,10}$";varmatch =Regex. Match (phone. ToString (), regex);

returnmatch.Success;
}
}

```

```
}).Select(phone => phone.ToString()).ToArray();
```

```
// СБШ:
```

```
messages = lastCalls.OfType<object>().Select(o => o.ToString()).ToArray();
```

```
}
```

```
else
```

```
{
```

```
Console.WriteLine("файл не містить телефонних номерів або повідомлень");
```

```
}
```

```
    }
```

```
#endregion
```

```
}
```

```
}
```

Додаток Е

Лістинг програми обробки зворотного зв'язку

```
namespaceListener
{
usingSystem;using
System.IO;
usingSystem.IO.Ports;usingSystem.Reflecti
on;usingSystem.Text;usingSystem.Text.Reg
ularExpressions;
usingExcel = Microsoft.Office.Interop.Excel;

classProgram
{
///<summary>
///the port.
///</summary>
privatereadonlySerialPortport =newSerialPort();

///<summary>
///The row number.
///</summary>privatereadonlyintrowNum
ber = 1;
privatereadonlyintcolumnNumber = 1;

staticvoidMain(string[] args)
{
if(OpenPort())
{
```

```

// Запускаємо Excel.
varexcel =newExcel.Application();

// Отримаємо нову робочу книгу.
varworkBook =
excel.Workbooks.Add(string.Empty);varoSheet =
(Excel._Worksheet)workBook.ActiveSheet;
AddHeader(oSheet);
port.DataReceived
+=newSerialDataReceivedEventHandler((objectsender,SerialDataReceivedE
ventArgse) =>
{
SerialPortsp =
(SerialPort)sender;stringindata =
sp.ReadExisting();

if(indata.Contains("Tel:"))
{
rowNumber++;
columnNumber = 1;

vartest =Regex. Split
(indata,@"\D+");varindex =
indata.IndexOf('+');varstrLength =
indata.Length;
indata = indata.Substring(index, strLength - index);

varphoneNumber =Regex. Match (indata,@"\d+").Value;

oSheet.Cells[rowNumber, columnNumber].Value = phoneNumber;

```

```

columnNumber++;
}
elseif(indata.Contains("+DTMF:"))
{
// натиснута кнопка
varkey = indata.Replace("+DTMF: ",string.Empty).Substring(0, 1);

oSheet.Cells[rowNumber, columnNumber].Value = key;
columnNumber++;
}

Console.WriteLine("Server data Received:
");Console. Write (indata);
});

Console.WriteLine("Listening...");Console.Re
adKey();
// зберігаємо після натискання будь-якого
клавiшаSaveDoc(excel, workbook);
}
}

///<summary>
///The add header.
///</summary>
///<param name="oSheet">
///The o sheet.
///</param>privatetaticvoidAddHeader(Excel._Worksheet
oSheet)
{

```



```

// Додавати заголовки таблиць
осередків.oSheet.Cells[1, 1] ="Номер
телефону";

// Число
колонок.intcountColu
mn = 12;

for(inti = 1; i<= countColumn; i++)
{
oSheet.Cells[1, i + 1] =$"Команда {i}";
}

varlastColumnLetter = (char)(countColumn + 65);

// Format A1: {Last column} as bold, vertical alignment = center.
// countColumn(columncount) + 65
// Перетворимо int на тип char, щоб отримати реальний символ за кодом
ASCII
oSheet.Range["A1", $" {lastColumnLetter} 1"].Font.Bold =true;
oSheet.Range["A1", $" {lastColumnLetter} 1"].VerticalAlignment =
Excel.XlVAlign.xlVAlignCenter;

// AutoFit columns A: lastColumnLetter.
oSheet.Range["A1", $" {lastColumnLetter} 1"].EntireColumn.AutoFit();

}

///<summary>
////The save doc.

```

```

///</summary>
private static void SaveDoc (Excel.Application Excel, Excel.Workbook workbook)
{
    // локація папки
    string executableLocation =
    Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location);
    var file = $"log_{DateTime.Now:MM_dd_yy H mm}.xls";
    // локація xls файлу
    string xlsLocation = Path.Combine(executableLocation,
    file);
    if (!File.Exists(xlsLocation))
    {
        throw new InvalidOperationException(), file);
    }

    excel.Visible = true;
    excel.UserControl = false;

    workbook.SaveAs(
        xlsLocation,
        Excel.XlFileFormat.xlWorkbookDefault, Type
        .Missing,
        Type.Missing, false,
        false,
        Excel.XlSaveAsAccessMode.xlNoChange, Type
        .Missing,
        Type.Missing, Type
        .Missing, Type
        .Missing);

    workbook.Close();
}

```

```
excel.Quit();
}

///<summary>
///The Open port.
///</summary>
///<returns>
///The<see cref="bool"/>.
///</returns>privatestaticbool
OpenPort()
{
try
{
port.BaudRate = 9600;
port.DataBits = 8;

port.ReadTimeout = 5000;
port.WriteTimeout = 2500;

port.Encoding =Encoding. GetEncoding("windows-1251");
port.PortName = "COM8";
if(port.IsOpen)
{
port.Close();
}

port.Open();r
eturntrue;
}
```

```
catch(Exception ex)
{
    Console.WriteLine("Не вдалося відкрити порт", ex.
    Message);return false;
}
}
}
}
```

Додаток Є

Лістинг програми контролера системи

```

#include<SoftwareSerial.h>#include<DFPlayer_Mini_Mp
3.h>
SoftwareSerial SIM900(8,9);          // 8 – RX Arduino (TX SIM900), 9 – TX
Arduino (RX SIM900)
SoftwareSerialmp3(10,11);          // 10 – RX Arduino (TX MP3), 11 –
TX Arduino (RX MP3)
String _response = "";           // Змінна для зберігання відповіді
модуляString number = "";
// String operator_number = "+79131188811"; // Номер чергового
оператораvoidsetup()
{
pinMode(13,OUTPUT);              // Індикація стану
пристроюSerial.begin(9600);      // Швидкість обміну даними з
комп'ютеромmp3.begin(9600);      // Швидкість обміну даними з mp3
програвачем
mp3_set_serial(mp3);              //Установка порту для керування mp3
програвачем
mp3_set_volume(2);               //Встановлення рівня
гучності програвача (мінімум)
// mp3_set_EQ(0);// EQ налаштування mp3 якщо потрібно????
Serial.println("Start!");
SIM900.begin(9600);              // Швидкість обміну даними з
модемомsendATCommand("AT",true); // Автоналаштування
швидкості
// sendATCommand("AT+CMGF=1",true); // УвімкнутиTextModeдля
SMSsendATCommand("AT+DDETECT=1",true); // Включаємо прийом DTMF
намодемодо
{

```

```

_response = sendATCommand("AT+CLIP=1",true);// Включаємо АОН
_response.trim();           // Забираємо пробільні символи на початку
та наприкінці
}while(_response != "OK");   // Не продовжувати роботу, поки
модем не поверне ОК

Serial.println("CLI in use"); // Модем працює...
// Тест та налаштування модему закінчені!
}
String sendATCommand(String cmd,boolwaiting)
{
String _resp = ""; // Змінна для зберігання
результатуSerial.println(cmd);// Дублюємо команду на
монітор порту
SIM900.println(cmd); // Відправляємо команду
модулюif(waiting)
{// Якщо потрібно дочекатися відповіді...
_resp =waitResponse(); // ... чекаємо, коли буде передана
відповідь
// Якщо EchoMode вимкнено (ATE0), то ці 3 рядки можна
закоментувати
if(_resp.startsWith(cmd))
{// Забираємо з відповіді команду, що дублюється.
_resp = _resp.substring(_resp.indexOf("r", cmd.length()) + 2);
}
Serial.println(_resp); // Дублюємо відповідь на монітор порту
}
return _resp; // Повертаємо результат. Порожньо, якщо є проблема
}

```

```

StringwaitResponse()
  { // Функція очікування відповіді та повернення отриманого
результатуString _resp =""; // Змінна для зберігання результату
long_timeout = millis() +10000; // Змінна для відстеження
таймууту (10 секунд)
while(! SIM900.available() &&millis() < _timeout) { };// Чекаємо на відповідь 10
секунд, якщо прийшла відповідь або настав тайм-аут, то...
if( SIM900.available())
  { // Якщо є, що зчитувати...
_resp = SIM900.readString(); // ... зчитуємо та запам'ятовуємо
  }
  else
  { // Якщо прийшов таймаут,
то...Serial.println("Timeout..."); // ... сповіщаємо про
це і...
  }
return_resp; // ... повертаємо результат. Пусто, якщо проблема
}
voidloop()
{
if( SIM900.available())
  { // Очікуємо на прихід даних (відповіді) від модему...
_response =waitResponse(); // Отримуємо відповідь модему
для аналізу
_response.trim();
Serial.println(">"+_response); // ...ідубльовані в Serial позначенням
">"if(_response.startsWith("RING"))
  { // Прийде дзвінку...
number = _response.substring(16, 28);// Висмикуємо символ із 7 позиції

```

```

довжиною 1 (по 8)
Serial.println("Tel: "+ Номер);
delay (2000);           // секундна
затримкаsendATCommand("ATA",true); // приймаємо вхідний дзвінок
delay(1000);
mp3_play(6);
}
// Тепер із даними можна
працюватиif(_response.startsWith("+D
TMF:"))
{// Якщо відповідь починається з "+DTMF:" тоді:
// Парсим отримана відповідь...
// Ми знаємо, що після цієї команди буде пробіл і один символ (натиснута
клавiша),
// отже можна відразу його " висмикувати " :
Stringsymbol = _response.substring(7, 8);// Висмикуємо символ із 7 позиції
довжиною 1 (по 8)
Serial.println("Key: "+ symbol);
// Додаємо логіку для отриманих
символівif(symbol == "1")
{
mp3_play(0);
}
elseif(symbol == "2")
{
mp3_play(2);
Serial.println("Feedback:"+ Номер);
}
elseif(symbol == "3")
{

```



```

mp3_play(3);
  Serial.println("New lead: "+ Homep);
}
elseif(symbol == "0")
{
mp3_stop();
/*sendATCommand("AT+CMGF=1r", true);
sendATCommand("AT+CMGS=\"\" + operator_number + "\"", true);
SIM900.println("TEST" + number + char(26));
delay (100); */
}
if(symbol == "4")
{
mp3_play(4);
}
}
elseif(_response.startsWith("NO"))
{// Поклали
трубкуmp3_stop();
number = ""; // Очищаємо змінну номер телефону
}
}
if(Serial.available())
{ // Очікуємо команди по Serial...
SIM900.write(Serial.read()); // ...і відправляємо отриману команду модему
};
}

```

Додаток Ж

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Апаратно-програмна система контролю та інформування учасників колективної роботи»

Тип роботи: _____ магістерська кваліфікаційна робота _____

Підрозділ _____ кафедра обчислювальної техніки, ФІТКІ _____

Показники звіту подібності Unicheck

Оригінальність 99% Схожість 1%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Захарченко С. М.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Годун В. В.

Керівник роботи



Азарова А. О.