ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра телекомунікацій, медійних та інтелектуальних технологій

КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

Бакалавр
Освітній рівень

Спеціальність 172 Телекомунікації та радіотехніка
Шифр і назва спеціальності

на тему Засіб автоматизованого захисту від несанкціонованого спостереження

КПТР.02051.01.03.ПЗ

Виконав:

студент 4 курсу, група ТР2-20-1 _____ О. О. Разовий
підпис ініціали, прізвище

Керівник: канд. техн. наук, доц. _____ В. С. Петрушак
підпис ініціали, прізвище

Нормоконтроль _____ О. С. Пивовар
підпис ініціали, прізвище

До захисту допускаю:

Зав. Кафедри: д-р техн. наук, порф. _____ С. К. Підченко
підпис ініціали, прізвище

10 06 2024р.

Хмельницький, 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*

Кафедра *телекомунікацій, медійних та інтелектуальних технологій*

Освітній рівень *бакалавр*

Галузь знань *17 «Електротехніка та телекомунікації»*
шифр і назва

Спеціальність *172 «Телекомунікації та радіотехніка»*
шифр і назва

Освітня програма *«Телекомунікації, медійні технології та інтелектуальні мережі»*

ЗАТВЕРДЖУЮ

Зав. Кафедрою ТМІТ

підпис, дата

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

Разовому Олександру Олеговичу
(Прізвище, ім'я, по батькові студента)

1 Тема проєкту: *Засіб автоматизованого захисту від несанкціонованого спостереження*
керівник проєкту *Петрушак Володимир Степанович, к.т.н., доцент*

Затверджено наказом ректора університету від «15» 02 2024р. №8

2 Строк подання студентом проєкту на кафедру: «1» 08 2024р.
3 Вихідні дані до проєкту 1. провести проєктування засобу автоматизованого захисту від несанкціонованого спостереження 2. ескіз схеми приймача
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити):
1. Огляд принципів функціонування аналогічних пристроїв 2. Розробка структури пристрою та вибір основних компонентів 3. Розробка схеми електричної принципової 4. Розробка алгоритму роботи та програмного забезпечення
5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
1. Схема електрична структурна 2. Схема електрична принципова 3. Схема алгоритму роботи

6 Консультанти розділів кваліфікаційного проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 01.03.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування виду роботи	Термін виконання етапів проєкту	Відмітка наукового керівника
1	Вибір тематики проєкту	Лютий	Виконано
2	Розробка завдання	Лютий	Виконано
3	Складання графіку	Лютий	Виконано
4	Огляд принципів функціонування аналогічних пристроїв	Лютий-березень	Виконано
5	Розробка структури пристрою та вибір основних компонентів	Березень	Виконано
6	Розробка схеми електричної принципової	Березень	Виконано
7	Розробка алгоритму роботи	Квітень	Виконано
8	Розробка програмного забезпечення	Квітень	Виконано
9	Розробка текстової частини	Квітень-травень	Виконано
10	Розробка графічної частини	Травень	Виконано
11	Остаточне коригування	Травень	Виконано
12	Нормоконтроль	Червень	Виконано
13	Підготовка до захисту	Червень	Виконано

Студент _____ О. О. Разовий
підпис ініціали, прізвище

Керівник проекту _____ В. С. Петрушак
підпис ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційного проекту:

«Засіб автоматизованого захисту від несанкціонованого спостереження»

Автор роботи: Разовий Олександр Олегович

Керівник роботи: канд.техн.наук, доц. Петрушак Володимир Степанович.

Пояснювальна записка: 60 сторінок, 1 таблицю, 32 рисунки, 26 джерел.

Графічна частина: 3 креслення, 9 презентаційних слайдів.

Ключові слова: спостереження, відеоспостереження, STM32, FPV.

Ця робота присвячена розробці засобу автоматизованого захисту від несанкціонованого відеоспостереження, який допоможе виявляти та нейтралізовувати загрози, пов'язані з незаконним використанням відеокамер, що передають зображення використовуючи радіозв'язок. Розглядаються сучасні методи та технології, що можуть бути використані для створення ефективної системи захисту. Метою кваліфікаційного проекту є розробка схем та програмного забезпечення для засобу протидії несанкціонованому відеоспостереженню.

Розд. Разовий О.О.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

RISC — Reduced Instruction Set Computing

SRAM — Static random access memory

ПЗ — програмне забезпечення

API — Application Programming Interface

GPIO - General-purpose input/output

UART — universal asynchronous receiver/transmitter

USART — universal synchronous and asynchronous receiver-transmitter

SPI — Serial Peripheral Interface

I^2C — Inter-Integrated Circuit

ADC — analog-to-digital converter

RTOS — Real-Time Operating System

ОП — Операційний підсилювач

РК — рідкокристалічний дисплей

ГКН — генератор керований напругою

№ рядка	Формат	Позначення	Найменування	Кількість	№ екземпл.	Примітка
1			Документація загальна			
2						
3	А4		Завдання на дипломний проєкт	1	1	
4						
5	А4		Анотація	1	1	Укр.
6						
7	А4	КПТР.020051.01.03.ПЗ	Засіб автоматизованого захисту			
8			від несанкціонованого спостереження			
9			Пояснювальна записка	1	1	60 арк.
10						
11	А4	КПТР.020051.01.03.ПЕ3	Засіб автоматизованого захисту			
12			від несанкціонованого спостереження			
13			Перелік елементів	1	1	3 арк.
14						
15			Документація графічна			
16						
17	А3	КПТР.020051.01.03.Е1	Засіб автоматизованого захисту			
18			від несанкціонованого спостереження			
19			Схема електрична структурна	1	1	1 арк.
20						
21	А3	КПТР.020051.01.03.Е3	Засіб автоматизованого захисту			
22			від несанкціонованого спостереження			
23			Схема електрична принципова	1	1	3 арк.
24						
25	А3	КПТР.020051.01.03.СА	Засіб автоматизованого захисту			
26			від несанкціонованого спостереження			
27			Схема алгоритму	1	1	1 арк.

КПТР.020051.01.03.ВП

Зм	Лист	№ докум	Підпис	Дата
Розробив		Разовий О. О.		
Перевірив		Петрушак В. С.		
Н.контр		Пивовар О. С.		
Затверд.		Підченко С. К.		

Засіб автоматизованого захисту від несанкціонованого спостереження

Відомість проекту

Літ.	Аркуш	Аркушів
н	1	1

ХНУ, ФІТ

ЗМІСТ

ВСТУП

Активний розвиток засобів відеоспостереження призвів до важливої потреби в захисті від їх неправомірного використання. Системи захисту від несанкціонованого відеоспостереження стали важливим компонентом безпеки у різних галузях, включаючи військовий, цивільний та комерційний сектори.

Незважаючи на численні переваги відеоспостереження, такі як підвищення рівня безпеки та можливість моніторингу, існують серйозні ризики, пов'язані з порушенням приватності та конфіденційності. Несанкціонований доступ до відеозаписів може призвести до витоку чутливої інформації, що становить загрозу для особистої та національної безпеки. Тому необхідність у розробці та впровадженні ефективних засобів автоматизованого захисту від несанкціонованого відеоспостереження стає все більш актуальною.

Сучасні технології пропонують різні підходи до захисту від несанкціонованого відеоспостереження, включаючи використання шифрування даних, технології блокування сигналів, системи виявлення та знешкодження шпигунських камер. Крім того, розвиток штучного інтелекту та машинного навчання відкриває нові можливості для створення інтелектуальних систем захисту, здатних автоматично виявляти і реагувати на загрози у режимі реального часу.

Використання засобів автоматизованого захисту від несанкціонованого відеоспостереження у військовій сфері є критично важливим для забезпечення безпеки та конфіденційності військових операцій. Ці засоби включають різні технології та методи, які дозволяють виявляти, блокувати та нейтралізувати загрози, пов'язані з відеоспостереженням.

У теперішній час для ведення несанкціонованого відеоспостереження дуже часто використовуються дрони, тому важливо мати засоби для їх виявлення та нейтралізації. Це можуть бути системи радіочастотного глушіння, технології кібер-атак для взяття під контроль дронів або фізичне збивання дронів за допомогою спеціалізованих пристроїв.

Можна використовувати спеціальні пристрої, які створюють оптичні або радіочастотні перешкоди, що робить відеозаписи неякісними або неможливими для аналізу. Наприклад, лазерні пристрої можуть засліплювати камери, а радіочастотні перешкоди можуть блокувати передачу даних від камер.

Автоматизовані системи можуть виявляти наявність камер відеоспостереження, включаючи приховані або масковані пристрої. Це досягається шляхом використання технологій виявлення інфрачервоного випромінювання, радіочастотних сигналів або магнітних полів. Після виявлення камери система може активувати засоби глушіння, що блокує сигнал від камери до приймача.

РОЗДІЛ 1 ОГЛЯД ПРИНЦИПІВ ФУНКЦІОНУВАННЯ АНАЛОГІЧНИХ ПРИСТРОЇВ ТА СФЕР ЇХ ЗАСТОСУВАННЯ

1.1 Принципи функціонування

Системи захисту від несанкціонованого відеоспостереження базуються на різноманітних технологіях для виявлення, ідентифікації та нейтралізації небажаних пристроїв. Серед основних принципів функціонування можна виділити[1]:

- Радіотехнічне виявлення. Для виявлення сигналів від FPV-дронів та їх подальшої ідентифікації використовуються радари та інші радіотехнічне системи.
- Акустичне виявлення. Для виявлення використовуються акустичні сигнали, що випромінюються дронами.
- Візуальне виявлення. Для виявлення та відстеження засобів, що здійснюють спостереження застосовуються відеокамери та системи обробки зображень на базі штучного інтелекту.
- Системи радіо-перехоплення, що забезпечують можливість перехоплення та блокування радіо-сигналів, що керують дронами.
- Фізичне знищення за допомогою спеціальних рушниць та інших засобів.

Найчастіше для виявлення дронів використовується саме радіотехнічне виявлення, тому що таким способом можна виявити літальний апарат на досить великій відстані. Для виявлення дронів використовуються різноманітні технології, включаючи радари та інші радіочастотні системи(Рис. 1.1). Але не зважаючи на те, що цей метод є найпопулярнішим, з розвитком дронів використовувати його стає все складніше. Наприклад, виробники дронів додають можливість змінювати частоту зі на якій відбувається передача сигналу між

літальним апаратом та пультом керування, що робить старі системи радіотехнічного виявлення неефективними.

Рисунок 1.1 — Спеціальні пристрої для виявлення дронів

Як бачимо з таблиці 1.1, сучасні засоби виявлення дронів мають забезпечувати покриття діапазону від 850МГц до 6ГГц. Ще однією проблемою є використання спеціальних ретрансляторів сигналу, що можуть значно збільшити дальність польоту FPV-дрона.

Таблиця 1.1 — Найпоширеніші частоти дронів

Частота	Виробник/Назва
860-930МГц (900M)	FPV дрони «Express LRS» — сигнал керування FPV дрони «CrossFire» — сигнал керування FPV дрони «FrSky» — сигнал керування
1100-1300МГц (1.2G)	Дальнобійні FPV передавачі — відеосигнал Комерційні дрони (DJI, Autel та інші) — GPS L2/L5[2]
1560-1610МГц (1.5G)	Усі дрони з GPS

Продовження таблиці 1.1

2400-2500МГц (2.4G)	DJI/Autel — відеосигнал Приймач FrSky ACCST — відеосигнал DJI FPV Remote Control — сигнал керування BetaFPV Cetus FPV Kit — сигнал керування
5150-5250МГц (5.2G)	AUTEL Lite — сигнал керування AUTEL EVO III — сигнал керування AUTEL EVO Max 4T — сигнал керування
5725-5850МГц (5.8G)	DJI (Mavic, Air, Mini, Matrice) — відеосигнал Autel — відеосигнал

До радіотехнічного виявлення дронів відноситься виявлення за допомогою радарів. Радари для виявлення дронів працюють на принципі відбиття радіосигналу від об'єкта (у цьому випадку дрона) і аналізу отриманих даних. Різноманітні типи радарів, включаючи пасивні та активні, можуть бути використані для виявлення дронів. Використання радіолокації дає максимально повну інформацію про розміри дрона, напрямок руху та відстань до нього.

Популярним способом виявлення FPV-дронів є виявлення сигналів керування чи відеосигналу. Шляхом виявлення цих сигналів можна встановити наявність дронів у певній області. Поки що цей метод працює добре, але в наш час вже ведеться розробка дронів зі штучним інтелектом, які можуть слідкувати за об’єктом в автоматичному режимі, тому їм не потрібен постійний зв’язок з пультом керування[4].

Також можуть використовуватись системи ідентифікації сигналу дрона. Ці системи використовують радіочастотні сигнали, що випромінюються дронами, для того щоб розпізнавати типи дронів, їх розташування та інші характеристики.

Ці технології можуть використовуватися як незалежно, так і в поєднанні одна з одною для забезпечення надійного виявлення дронів у різних умовах та середовищах. Більшість сучасних систем також мають можливість ідентифікації дронів, що дозволяє операторам аналізувати та реагувати на потенційні загрози.

Із розвитком штучного інтелекту візуальне виявлення дронів здобуває все більше значення в контексті забезпечення безпеки та контролю над повітряним простором. Використання відеокамер та систем обробки зображень на базі штучного інтелекту може значно полегшити виявлення та відстеження дронів. Для спостереження можуть використовуватись як звичайні відеокамери так і інфрачервона зйомка, оскільки деякі дрони може бути важко виявити за допомогою звичайних камер, але їхні теплові сліди можна виявити за допомогою інфрачервоних (теплових) камер.

Разом з використанням камер використовуються різні системи обробки зображень, такі як комп'ютерний зір, штучний інтелект та додатково системи виявлення руху. Найпоширенішою системою зі штучним інтелектом, що використовується для виявлення дронів є програмне забезпечення DroneTracker. Ця система може розрізняти дрони різних типів, а також відрізняти їх від інших рухомих об’єктів, наприклад літаків чи птахів. Система DroneTracker використовує для виявлення те лише відеокамери, а і інші способи виявлення, такі як акустичне та радіотехнічне виявлення[3]. На рисунку 1.2 зображені спеціальні пристрої для роботи з DroneTracker.

Рисунок 1.2 — Апаратне забезпечення для системи DroneTracker

Штучний інтелект DroneTracker навчений на мільйонах зображень та використовує базу даних, яка є найбільш актуальною та повною на даний час.

Якщо ж потрібно реалізувати свою систему виявлення дронів зі штучним інтелектом, то можна використати популярні нейронні мережі для аналізу зображень, такі як Faster R-CNN, Mask R-CNN, SSD, YOLO.

Системи візуального виявлення зазвичай інтегровані з іншими системами для підвищення точності:

1. Геолокація та слідкування:
   - GPS: Використання GPS для визначення місця розташування дронів.
   - Системи слідкування: Встановлення систем слідкування, які можуть автоматично націлювати камеру на дрона та відстежувати його рух.
2. Інтеграція з іншими технологіями:
   - Радарні системи: Інтеграція з радарними системами може поліпшити ефективність виявлення дронів, особливо у важких погодних умовах
   - Системи радіо-перехоплення, за допомогою яких можна перешкоджати прийому відеосигналів, сигналів що керують дронами, сигналів GPS.
   - Фізичне знищення за допомогою спеціальних спеціальних засобів, таких як антидронові рушниці.

Всі ці компоненти можуть бути інтегровані в систему виявлення та відстеження дронів, яка дозволить ефективно контролювати повітряний простір та захищати важливі об'єкти від можливих загроз.

Одним з можливих способів протидії FPV-дронам є перехоплення та блокування сигналів дронів яке стає все більш актуальною задачею, тому що фізично знищити дрон не завжди вдається через його швидкість та малі розміри. Розглянемо деякі технічні засоби, які можуть використовуватися для перехоплення та блокування радіо-сигналів, які керують дронами:

- Імпульсні завади: Цей метод включає в себе надсилання радіошумів на тому ж частотному діапазоні, на якому працює дрон. Це може призвести до тимчасового втрати зв'язку та контролю над дроном. Однак використання імпульсних завад може супроводжуватися проблемами, такими як перешкоджання для інших комунікаційних систем. Захист від фальшивих сигналів та електромагнітних перешкод є критичним елементом антидронових систем. Забезпечення високого рівня відповідності та захисту перехоплення даних може мінімізувати можливість обхідних шляхів атаки. Можлива структура технічного засобу для протидії дронам зображена на рисунку 1.3.

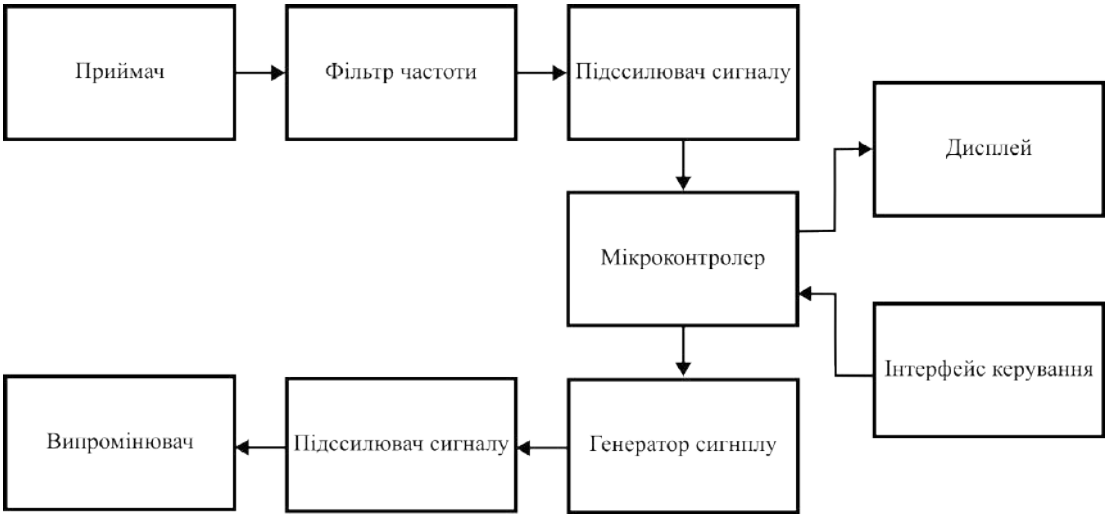


Рисунок 1.3 — Структура засобу захисту від відеоспостереження за допомогою генерації радіошумів

- Перехоплення та витягання даних (Sniffing): Використання антен та приймачів для перехоплення сигналів дронів. Після цього можна спробувати аналізувати та розшифровувати команди для дрона. Цей підхід може вимагати великих зусиль для аналізу інформації та взаємодії з різними типами дронів.
- GPS-перехоплення: Захоплення сигналів GPS, які використовуються для навігації дронів. Це може призвести до того, що дрон втрачає точність місця знаходження та не може коректно виконувати завдання.

1.2 Застосування систем захисту від несанкціонованого відеоспостереження

- Цивільне застосування: В аеропортах, енергетичних об'єктах та інших критичних інфраструктурних об'єктах антидронові системи служать для захисту від можливих загроз.
- Військове застосування: Антидронові системи відіграють важливу роль у забезпеченні безпеки військових об'єктів, перешкоджаючи використанню дронів противниками.
- Комерційне використання: Організації та підприємства використовують антидронові системи для захисту конфіденційної інформації, безпеки заходів та запобігання шпигунству.

Антидронові системи використовуються в цивільних сферах для захисту від можливих загроз, пов'язаних з безпілотними літальними апаратами (БПЛА або дронами). Ці системи можуть мати різноманітні функції та можливості, спрямовані на виявлення, ідентифікацію та нейтралізацію неправомірних дронів.

Безпека на заходах та подіях: Великі громадські заходи, такі як фестивалі, концерти, спортивні події чи політичні мітинги, можуть стати об'єктом

неправомірного використання дронів. Антидронові системи можуть допомагати виявляти та врегулювати доступ дронів в таких місцях.

Захист критичних інфраструктур: Антидронові системи можуть бути використані для захисту критичних об'єктів, таких як аеропорти, електростанції, промислові об'єкти та інші важливі інфраструктурні об'єкти, від можливих загроз дронів.

Контроль над територією: Антидронові системи можуть застосовуватися для нагляду та контролю над певною територією, забезпечуючи безпеку на об'єктах важливих для громадськості чи бізнесу.

Захист приватності: В ряді випадків дрони можуть порушувати приватність громадян, фільмуючи або фотографуючи заборонені області. Антидронові системи можуть допомагати у виявленні та обмеженні таких порушень.

Застосування в рятувальних операціях: Антидронові технології можуть бути використані в рятувальних операціях, де дрони можуть заважати чи навіть становити загрозу для рятувальників чи постраждалих.

Комерційне використання антидронових систем стало актуальним з ростом популярності та доступності безпілотних літальних апаратів (дронів). Антидронові системи призначені для виявлення, ідентифікації та втручання в роботу непридатних дронів, які можуть становити загрозу безпеці або конфіденційності.

Комерційні сценарії використання антидронових технологій включають:

Захист просторів: Комерційні підприємства, особливо ті, що працюють в чутливих галузях, таких як енергетика, транспорт, виробництво та інші, можуть використовувати антидронові системи для захисту своїх просторів від засобів відеоспостереження, які можуть порушити роботу або становити загрозу безпеці.

Події та заходи: Організатори масових заходів, таких як концерти, фестивалі, спортивні події та інші, можуть використовувати антидронові системи для запобігання неправомірному використанню дронів та забезпечення безпеки учасників і гостей.

Об'єкти інфраструктури: Комерційні об'єкти інфраструктури, такі як аеропорти, порти, нафтопереробні заводи та інші, можуть використовувати антидронові технології для захисту від можливих загроз.

Комплексне забезпечення захисту: Деякі компанії спеціалізуються на наданні комплексних рішень з безпеки, які включають в себе не лише антидронові системи, але і інші технології, такі як системи відеоспостереження, доступу та ідентифікації.

1.3. Перспективи розвитку

З великим розширенням використання дронів, виникають нові виклики для антидронових систем. Розробники стикаються з завданням вдосконалення технологій виявлення та нейтралізації, а також забезпечення сумісності з іншими системами безпеки. Перспективи розвитку включають в себе вдосконалення алгоритмів штучного інтелекту для автоматизованого управління антидроновими системами та використання новітніх технологій, для більш ефективного виявлення та нейтралізації дронів.

Одним із ключових напрямків розвитку є вдосконалення технологій виявлення дронів. Це включає застосування радарів, оптичних сенсорів, акустичних детекторів та систем радіочастотного моніторингу. Новітні розробки зосереджуються на створенні інтегрованих систем, які поєднують кілька типів сенсорів для забезпечення більшої точності та швидкості виявлення дронів. Важливу роль відіграє також використання штучного інтелекту для аналізу отриманих даних і виявлення загроз у режимі реального часу.

Другий напрямок розвитку пов'язаний з методами нейтралізації дронів. Тут виділяються технології радіоелектронної боротьби, що дозволяють блокувати або перехоплювати сигнали управління дронами. Крім того, активно розробляються і впроваджуються методи фізичного знешкодження, такі як використання спеціальних сіток, лазерних систем або навіть високошвидкісних дронів-

перехоплювачів. Серед перспективних рішень також розглядаються можливості впливу на навігаційні системи дронів, що дозволяє порушити їх здатність орієнтуватися в просторі.

Окремо варто зазначити розвиток антидронових систем для цивільних об'єктів. Це включає захист аеропортів, важливих інфраструктурних об'єктів, а також місць масового скупчення людей. В цьому контексті, зростає важливість розробки систем, які не тільки ефективні, але й безпечні для навколишнього середовища і людей[6]. Наприклад, технології, які мінімізують ризики падіння нейтралізованих дронів на територію з великою кількістю людей.

Суттєвий вплив на розвиток антидронових систем має також нормативно-правова база. Законодавчі ініціативи, що регулюють використання дронів та антидронових технологій, сприяють формуванню безпечного середовища для їх впровадження. Важливо, щоб ці ініціативи враховували як потреби безпеки, так і права громадян.

РОЗДІЛ 2 РОЗРОБКА СТРУКТУРИ ПРИСТРОЮ ТА ВИБІР ОСНОВНИХ КОМПОНЕНТІВ

2.1 Розробка структури пристрою

Основними вимогами до пристрою є:

- пристрій має бути розділений на дві частини: блок прийому сигналів та генерації шуму і блок керування, які можуть знаходитись один від одного на відстані кількох метрів;
- можливість визначити з якої сторони знаходиться засіб, що виконує відеоспостереження;
- живлення напругою 12В, що дозволить підключати живлення пристрою від бортової мережі автомобіля, чи акумулятора;
- можливість ручного шуму запуску;
- відображення значення сигналів та статусу генерації шуму на дисплеї;

Відповідно до цих вимог, у конструкції пристрою мають бути чотири приймачі, що підсилюють сигнал, до яких підключені перпендикулярно розташовані антени. Таким чином, можна буде визначити з якої сторони сила сигналу є найбільшою. Далі, підсилені сигнали потрапляють на входи блоку прийому та генерації шуму, роль якого виконує мікроконтролер. Цей мікроконтролер в залежності від рівня вхідних аналогових сигналів приймає рішення чи потрібно запустити або зупинити генерацію шуму, а також надсилає інформацію про рівень вхідних сигналів та статус генерації на блок керування. В свою чергу, блок керування приймає цю інформацію, та відображає її на дисплеї у текстовому вигляді. Також до блоку керування під’єднана кнопка, за допомогою якої оператор зможе вручну запустити генерацію шуму у разі необхідності. В такому випадку, блок керування надсилає інформацію до блоку прийому та генерації шуму, що потрібно запустити генерацію незважаючи на рівень сигналу.

Також потрібно подбати про надійність зв’язку між двома блоками пристрою, оскільки вони, відповідно до вимог, можуть знаходитись на відстані кількох метрів, а сигнал з мікроконтролера є занадто слабким для передачі даних на таку відстань. Для вирішення цієї проблеми можна використати спеціальні трансмітери, які будуть посередниками у з’єднанні двох мікроконтролерів, та забезпечать хороший зв’язок. Описана структурна схема зображена на рисунку 2.1.

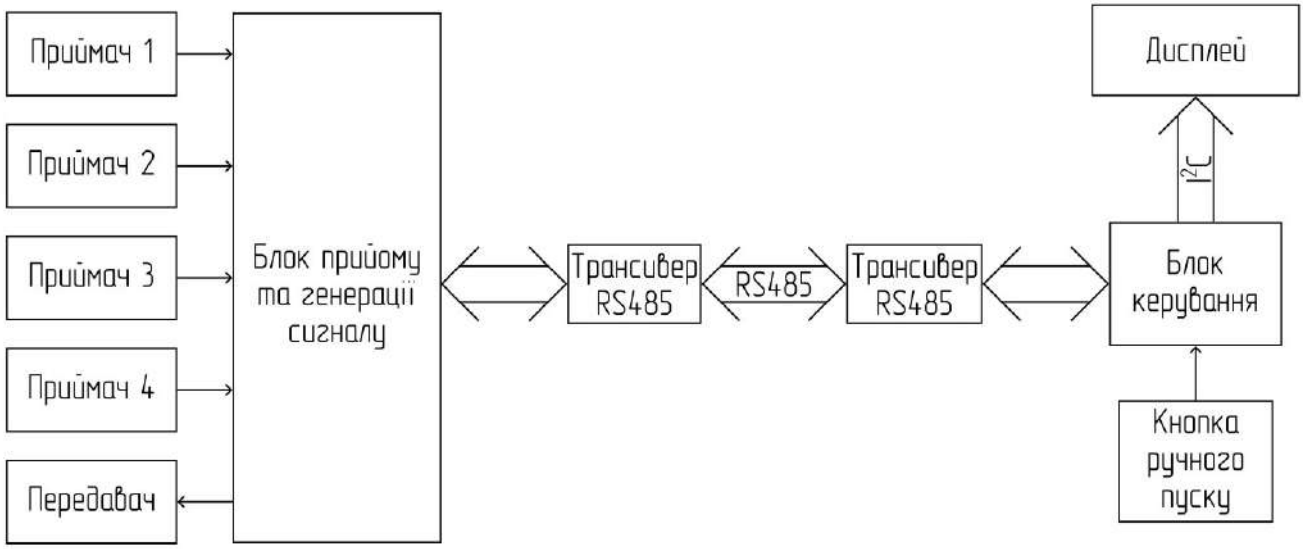


Рисунок 2.1 — Структурна схема пристрою

2.2 Вибір основних компонентів

У кожному з блоків буде використовуватись мікроконтролер STM32F030K6T6, який є частиною серії STM32F0 від STMicroelectronics і базується на ядрі ARM Cortex-M0.

Його основні характеристики:

- Ядро:
  - ARM Cortex-M0, 32-бітний RISC процесор;
  - Тактова частота до 48 МГц;

- Пам'ять:
  - 32 КБ флеш-пам'яті;
  - 4 КБ оперативної пам’яті (SRAM);
- Периферія:
  - 12-бітний АЦП з 10 каналами;
  - 16-бітний таймер з розширеними можливостями;
  - Чотири 16-бітні таймери загального призначення;
  - Незалежний сторожовий таймер (watchdog);
  - Системний таймер (SysTick) для операційних систем реального часу;
  - 26 універсальних входів/виходів;
- Комунікаційні інтерфейси:
  - USART: 1 канал;
  - SPI: 1 канал;
  - I^2C: 1 канал;
- Керування енергоспоживанням:
  - Робоча напруга: від 2.4 В до 3.6 В;
  - Три режими низького енергоспоживання: Sleep, Stop і Standby;
- Тактова частота:
  - Внутрішній RC осцилятор на 8 МГц з точністю ±1%;
  - Фазове автопідлаштування частоти (PLL) для генерації високих тактових частот;
  - Можливість підключення зовнішнього кварцового резонатора до 32 МГц;
- Корпус LQFP32 (7x7 мм) (Рис. 2.2);
- Інші функції:
  - Контролер прямого доступу до пам’яті (DMA), що дозволяє передавати дані від периферійних пристроїв напряму в оперативну пам’ять;

Ці характеристики роблять STM32F030K6 придатним для широкого спектра застосувань, включаючи промислову автоматизацію, споживчу

електроніку, та інші вбудовані системи, де потрібна низька вартість та ефективність. Також великим плюсом є те, що на деякі виводи мікроконтролера, незважаючи на те, що він живиться напругою 3,3В, можна подавати напругу величиною до 5В. Це дозволяє не використовувати спеціальних перетворювачів рівню сигналу з 5В до 3,3В, при підключенні деяких периферійних пристроїв. Ціна цього мікроконтролера на момент написання проекту становить 40-55грн.

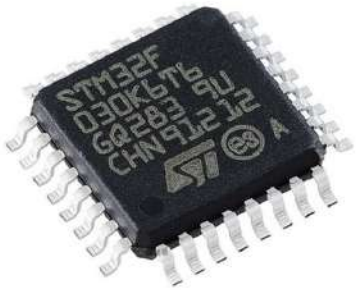


Рисунок 2.2 — Мікроконтролер STM32F030K6T6 у корпусі LQFP32

Оскільки потрібно відображати дані в текстовому вигляді, то для цього доцільно використати символьний РК дисплей. Розширення 16 на 2 символи буде достатньо для відображення чотирьох значень сигналів. Також краще використовувати такий дисплей з I2C модулем розширення виводів на мікросхемі PCF8574T, щоб не займати багато виводів мікроконтролера. Приклад такого дисплею зображений на рисунку 2.3.

Рисунок 2.3 — РК дисплей з модулем I^2C

Оскільки сигнали на виходах мікроконтролера занадто слабкі для забезпечення надійного зв’язку між двома блоками пристрою на відстані кількох метрів, для передачі даних використаємо мікросхему MAX3485.

Мікросхема MAX3485 — це низькопотужний високошвидкісний трансивер RS-485/RS-422, розроблений компанією Maxim Integrated. Вона використовується для забезпечення надійного та швидкого обміну даними в промислових системах, мережах, а також в інших застосуваннях, де потрібні стабільні комунікаційні лінії на великі відстані.

Основні характеристики MAX3485:

1. Напруга живлення: 3.3 В
2. Максимальна швидкість передачі даних: 10 Мбіт/с
3. Діапазон робочих температур: від -40°C до +85°C
4. Режим роботи: напівдуплекс
5. Захист від електромагнітних завад
6. Тип корпусу: SOP-8 або TDFN-8

Мікросхема MAX3491 функціонує як трансивер, що означає, що вона може працювати як передавач, і як приймач. Основні функції мікросхеми включають передачу та прийом диференціальних сигналів, що дозволяє зменшити вплив електромагнітних завад та покращити якість сигналу на великих відстанях.

Режим передавача активується, коли подається логічна одиниця на керуючий вхід DE (Driver Enable). У режимі передавача сигнал з входу DI (Driver Input) транслюється у диференційний сигнал на виводи A і B. Якщо на вході DI логічний нуль, то на виході A буде буде логічний нуль, а на виході B логічна одиниця. Якщо ж на вході DI буде логічна одиниця, то на виході A буде логічна одиниця, а виході B логічний нуль.

Режим приймача активується, коли сигнал на керуючому вході RE (Receiver Enable) знаходиться у стані логічного нуля. Приймач зчитує диференціальний сигнал з пари входів A і B та відтворює його у вигляді логічних рівнів на виході RO (Receiver Output). Типова схема підключення зображеня на рисунку 2.4.

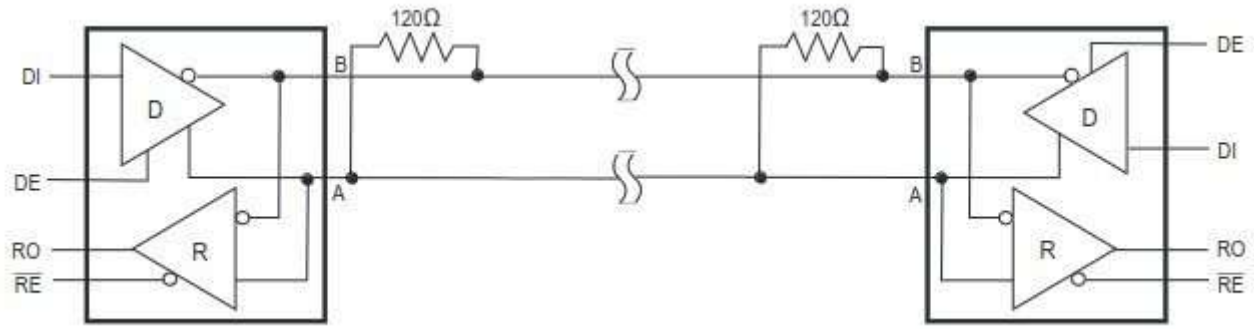


Рисунок 2.4 — З'єднання двох мікросхем MAX3485

MAX3485 має вбудований захист від електростатичних розрядів (ESD) до ±15 кВ, що забезпечує додаткову надійність в умовах промислового використання. Це досягається завдяки спеціальним захисним схемам на входах і виходах мікросхеми.

Мікросхема MAX3485 ідеально підходить для використання в різних промислових мережах, де важлива стабільна та швидка передача даних. Завдяки підтримці високої швидкості передачі до 10 Мбіт/с і широкому діапазону

робочих температур, вона є надійним рішенням для різноманітних застосувань, включаючи автоматизацію виробництва, контроль процесів, системи безпеки та інші.

РОЗДІЛ 3 РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ

3.1 Розробка схеми електричної принципової блоку керування

Основним елементом блоку керування є мікроконтролер. Мікроконтролеру для роботи потрібна певна мінімальна «обв'язка», наприклад, резистор, що підтягує сигнал скидання до напруги живлення чи кварцовий резонатор, що задає частоту роботи мікроконтролера.

У випадку з мікроконтролером STM32F030K6T6, нам потрібно під'єднати усі виводи VDD(живлення цифрової частини) та вивід VDDA (живлення аналогової частини) до джерела живлення +3,3В. Усі виводи VSS потрібно приєднати до загального провідника. Вивід скидання NRST потрібно під'єднати до живлення +3,3В, через резистор номіналом 10кОм. Вивід BOOT0, потрібно під'єднати до загального провідника, щоб мікроконтролер починав виконання програми з внутрішньої пам'яті. Також мікроконтролеру потрібен сигнал тактування, тому до виводів OSC_IN та OSC_OUT потрібно під'єднати кварцовий резонатор за схемою, зображеною на рисунку 3.1.

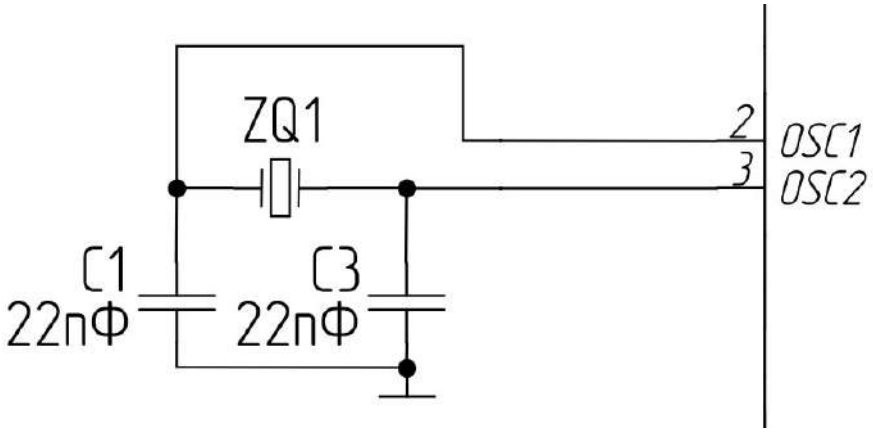


Рисунок 3.1 — Підключення кварцового резонатора до мікроконтролера

У даному пристрої використовується кнопка типу DTS-61K-V(Рис. 3.2) для запуску чи зупинки генерації шуму. Оскільки мікроконтролер STM32F030K6T6 має внутрішню схему підтяжки виводів до напруги живлення, яку можна увімкнути програмно, то додатковий резистор для кнопки не потрібен. Один

вивід кнопки підключаємо до виводу мікроконтролера, а інший до загального провідника. Схема підключення кнопки зображена на рисунку 3.3.

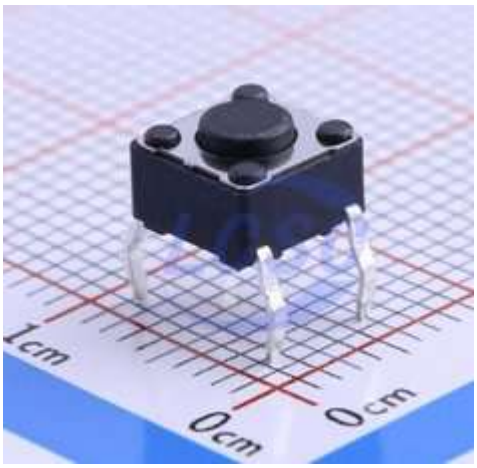


Рисунок 3.2 — Кнопка типу DTS-61K-V

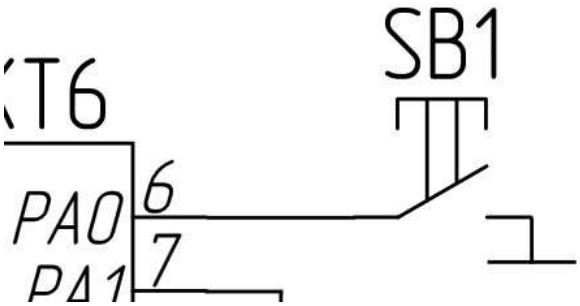


Рисунок 3.3 — Схема підключення кнопки до мікроконтролера

Для комунікації з блоком прийому сигналу та генерації шуму, відповідно до структурної схеми, використовується інтерфейс RS485. Роль трансивера, виконує мікросхема MAX3485. Її виводи DI та RO можна напряму під'єднати до виводів мікроконтролера, які можуть працювати з вбудованим в мікроконтролер драйвером UART. У мікроконтролера STM32F030K6T6 цими виводами є PA2 та PA3 або PA8 та PA9. Виводи PA8 та PA9, будуть використовуватись для передачі даних на дисплей, тому для комунікації з блоком прийому та генерації шуму використаємо PA2 та PA3. Також, мікросхемі MAX3485 потрібно вказати в якому режимі потрібно працювати в даний момент: як приймач або як передавач. Для цього мікросхема має виводи RE та DE, які можна об'єднати разом, тому що RE є

інверсним, та під’єднати їх до мікроконтролера. Також до мікросхеми MAX3485 потрібно приєднати живлення +3,3В і резистор між виводами A та B номіналом 120Ом, згідно з документацією. Виводи A та B потрібно вивести на роз’єм, тому що до них буде підключатись блок прийому сигналу та генерації шуму. В якості роз’єму можна використати WJ124-3.81-3P (Рис. 3.4). Схема підключення мікросхеми MAX3485 зображена на рисунку 3.5.

Рисунок 3.4 — Роз’єм для підключення блоків по RS485

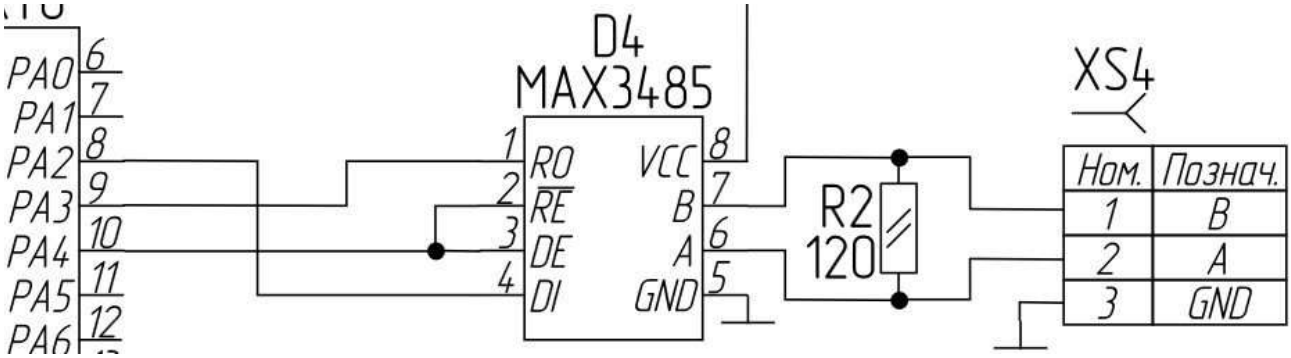


Рисунок 3.5 — Схема підключення мікросхеми MAX3485 до мікроконтролера

Для відображення даних в даному пристрої, використовується РК дисплей з I^2C модулем розширення виводів PCF8574T (Рис. 3.6). Для коректної роботи дисплею, йому потрібна напруга живлення +5В, тому і для передачі даних потрібно використовувати виводи що можуть працювати з напругою 5В та мають можливість працювати з вбудованим в мікроконтролер драйвером I^2C. У мікроконтролера STM32F030K6T6 цими виводами є PA8 та PA9, які можна програмно сконфігурувати як виводи SCL(тактування) та SDA(дані).

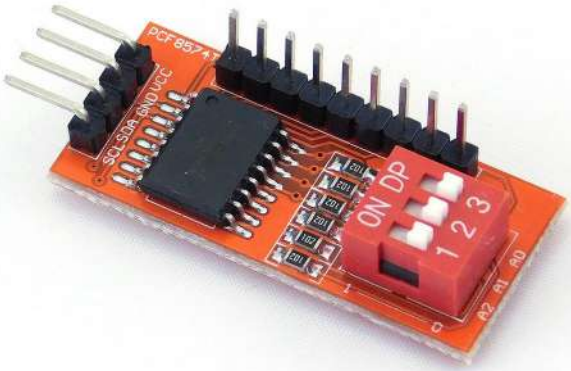


Рисунок 3.6 — модуль розширення виводів на мікросхемі PCF8574T

Для підключення цього дисплею можна використати роз'єм на 4 контакти типу B-2100S04P-A110 (Рис. 3.7), додатково зафіксувавши його за допомогою клею. Також варто додати роз'єм такого ж типу для прошивки мікроконтролера.

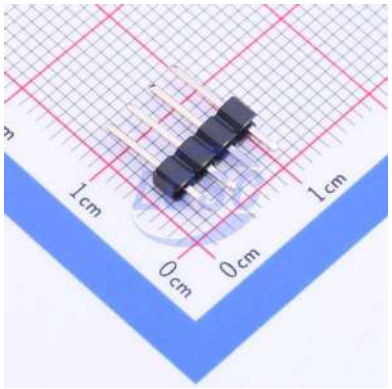


Рисунок 3.7 — Роз'єм для підключення дисплею

Для живлення цього блоку керування використаємо лінійні регулятори напруги серії AMS1117. Регулятори цієї серії мають наступні характеристики:

- Фіксовані вихідні напруги: 1.5В, 1.8В, 2.5В, 2.85В, 3.3В та 5.0В
- Максимальний вихідний струм: 1А;
- Максимальна вхідна напруга: 15В;
- Корпус: TO-252, SOT-223, 8L SOIC

Для живлення блоку використаємо регулятори на 5В та 3,3В в корпусі SOT-223. Згідно з документацією, на виході регуляторів слід встановити конденсатори ємністю 22мкФ. Роз’єм живлення використаємо типу WJ500V-5.08-2P-14-00A(Рис. 3.8).

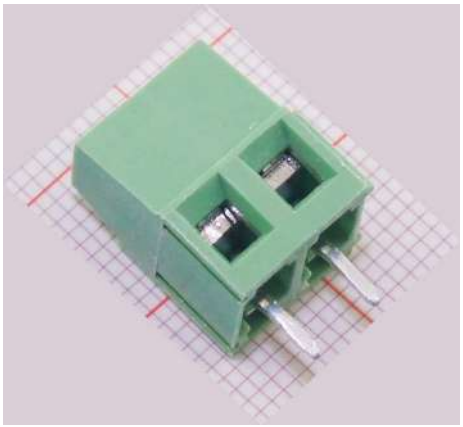


Рисунок 3.8 — Роз’єм типу WJ500V-5.08-2P-14-00A

В результаті проектування отримуємо схему блоку керування, зображену на рисунку 3.9.

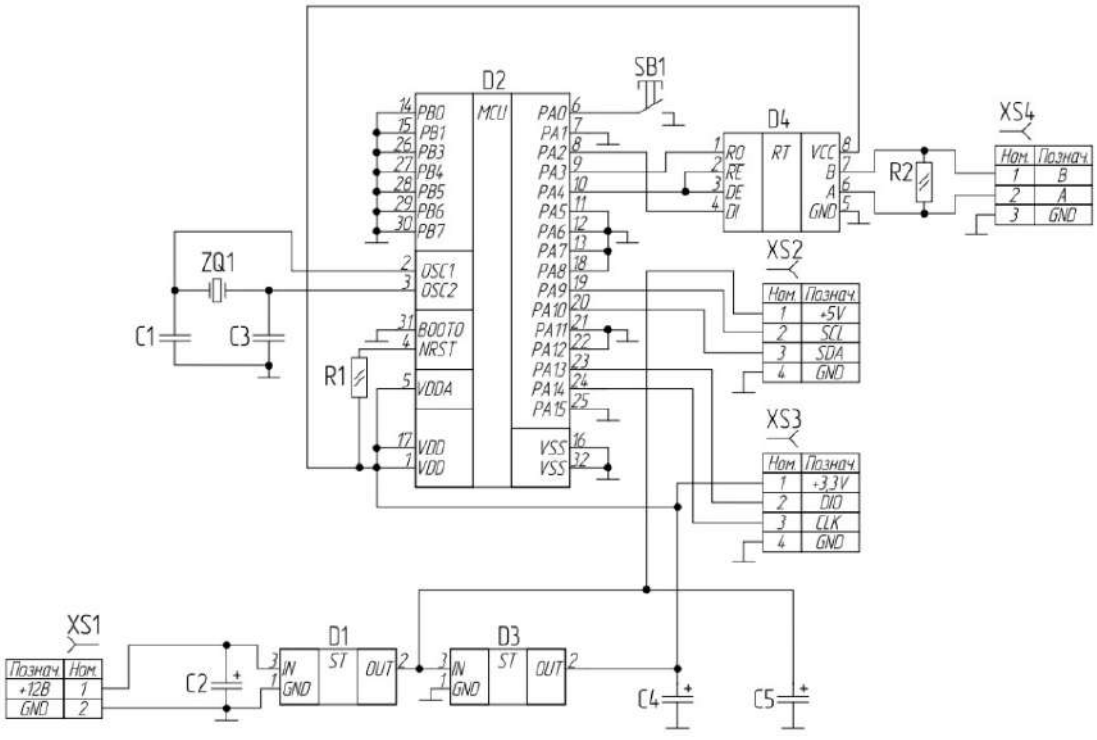


Рисунок 3.9 — Схема електрична принципова блоку керування

3.2 Розробка схеми електричної принципової блоку прийому та генерації сигналів

Для блоку прийому сигналу та генерації шуму використовується такий же мікроконтролер, як і для блоку керування, тому частина схеми буде аналогічною до схеми блоку керування.

Згідно зі структурною схемою, мікроконтролер у цьому блоці має аналізувати сигнал з чотирьох приймачів. Для цього потрібно під'єднати виводи, що можуть бути програмно сконфігуровані як входи АЦП, до роз'єму, через який будуть підключатись приймачі. У мікроконтролері STM32F030K6T6 можемо використати виводи PA1-PA4. Також на роз'єм необхідно вивести живлення +9В для приймачів. В якості роз'єму для для підключення приймачів використаємо роз'єм типу JST PH-6 (Рис. 3.10).

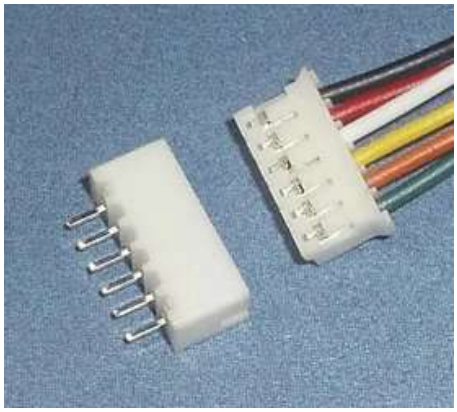


Рисунок 3.10 — Роз'єм типу JST PH-6

Для генерації шуму використаємо ГКН YSGM081008, який має наступні характеристики:

Номінальний діапазон налаштування: 870-960МГц;

Середня вихідна потужність: ≥7dBm;

Напруга живлення: 5В;

Діапазон напруги керування: 0-9В;

Робочий струм: 85мА;

Розміри: 7мм×9мм×2мм;

Рисунок 3.10 — Вигляд ГКН YSGM081008

Мікроконтролер буде формувати напругу керування ГКН за допомогою ШІМ сигналу, який згладжується конденсатором з номінальною ємністю 100мкФ, оскільки STM32F030K6T6 не має вбудованого цифро-аналогового перетворювача, щоб сформувати відповідну напругу. ШІМ сигнал буде генеруватись на виводі мікроконтролера PA6, тому цей вивід під'єднуємо до конденсатора та до виводу VT ГКН, який є входом для подачі напруги керування. На вивід VCC ГКН під'єднуємо напругу +5В, а вивід RFOUT підключаємо до роз'єму, через який сигнал буде потрапляти на підсилювач. В якості такого роз'єму використовується KH-SMA-K513-G(Рис. 3.11). Схема підключення ГКН зображена на рисунку 3.12.

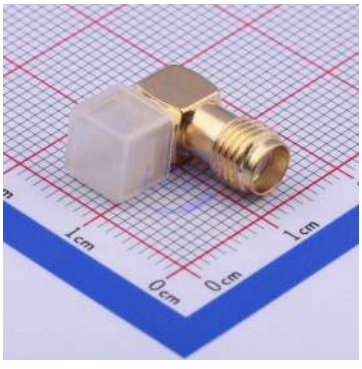


Рисунок 3.11 — Конектор типу KH-SMA-K513-G

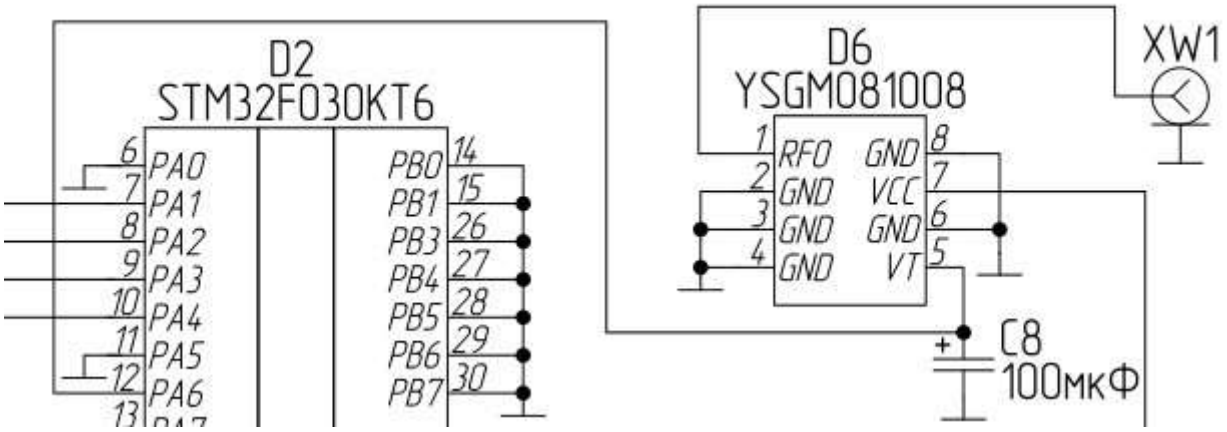


Рисунок 3.12 — Схема підключення ГКН до мікроконтролера

У блоці прийому сигналу та генерації шуму, крім напруг 3,3В та 5В, потрібна напруга 9В для живлення приймачів. Для утворення цієї напруги використаємо лінійний регулятор L7809. Цей регулятор має наступні характеристики:

- Вихідна напруга: 9В;
- Мінімальна вхідна напруга: 11В;
- Максимальна вхідна напруга: 35В;
- Вихідний струм: 1.5А;
- Вбудований захист від перегріву та короткого замикання;
- Корпус: TO-220, TO-220FP, DPAK, D^2PAK;

Решта елементів, що не описані в цьому підрозділі, такі як кварцовий резонатор, роз’єми живлення та прошивки мікроконтролера є аналогічними до тих, що використані при розробці блоку керування. В результаті проектування отримуємо схему блоку керування, зображену на рисунку 3.13.

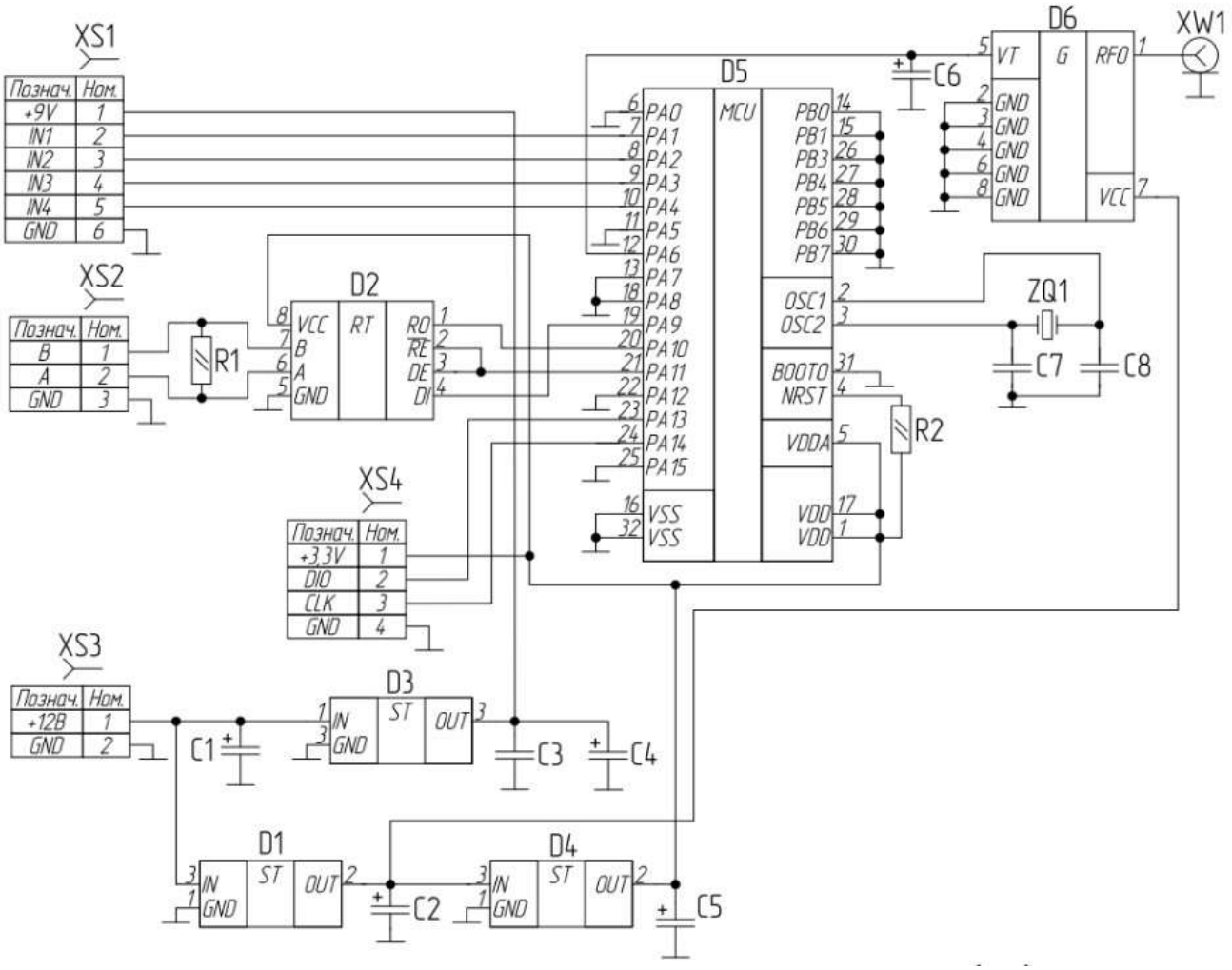


Рисунок 3.13 — Схема електрична принципова блоку прийому сигналу та генерації шуму

3.3. Розробка схеми електричної принципової приймача

Сема електрична принципова приймача розроблена на основі ескізу, який дано у завданні на кваліфікаційний проєкт (Рис. 3.14)., з деякими змінами.

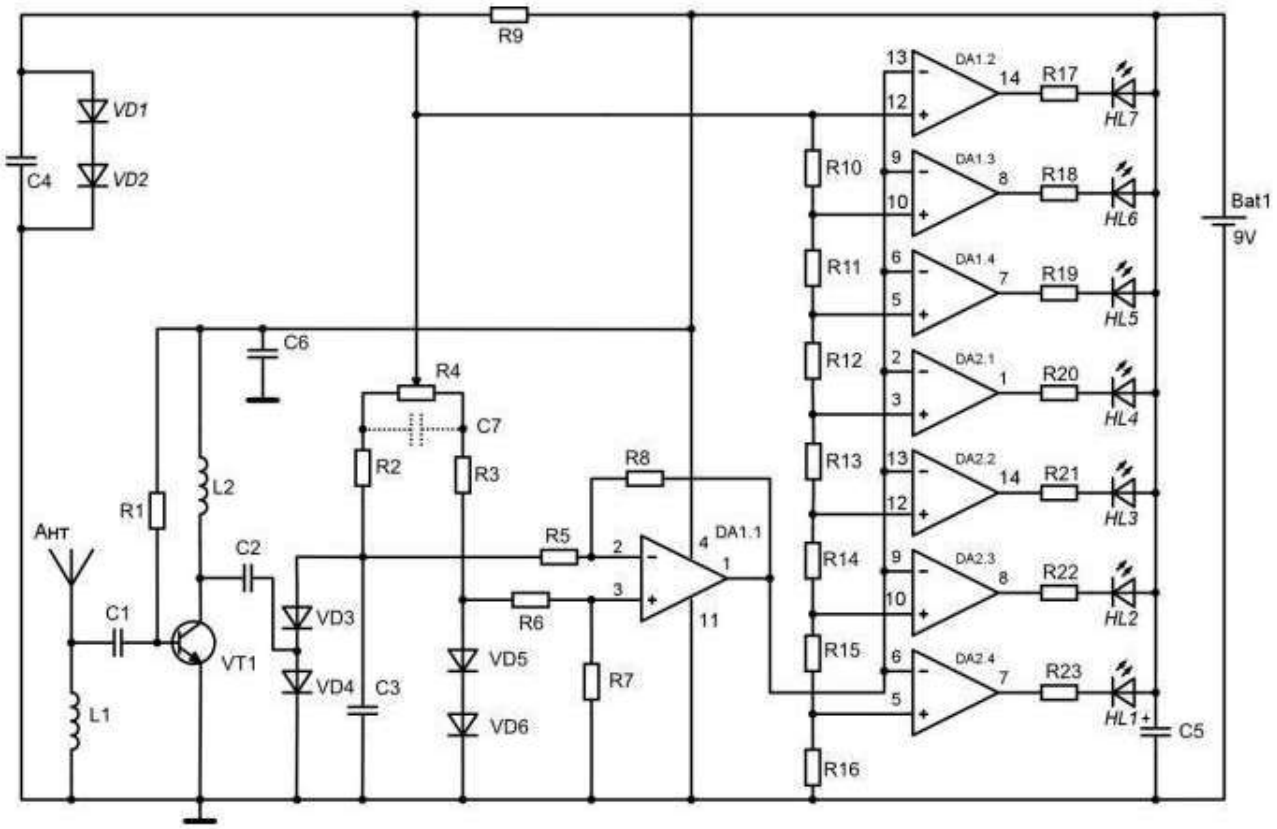


Рисунок 3.14 — Ескіз схеми приймача

В якості транзистора, що позначений на ескізі VT1 був використаний BFR92P, який призначений для використання у високочастотних схемах до 2ГГц, де потрібна висока швидкість перемикання та низький рівень шуму. Нижче наведено деякі його характеристики:

- Тип транзистора: NPN;
- Максимальна напруга колектор-емітер: 15В;
- Максимальний струм колектора: 45мА;
- Мінімальний коефіцієнт підсилення(hFE): 70;
- Корпус: SOT23;

В якості діодів використовуються 1N4148W. Це широко використовуваний діод загального призначення. Він відомий своїми швидкими перемикаючими характеристиками та надійністю. Ось його основні характеристики:

- Максимальна зворотна напруга: 75 В;

- Максимальний безперервний прямий струм: 300 мА;
- Піковий прямий імпульсний струм 2 А (не більше 1 мкс);
- Час відновлення: 4 нс;
- Ємність переходу 1,5 пФ;
- Корпус: SOD-123;

На ескізі, в якості операційних підсилювачів використовується мікросхема LM324, в корпусі якої знаходяться 4 операційні підсилювачі. Також на ескізі, на цих же підсилювачах зібраний АЦП. У даному проєкті потрібен лише один операційний підсилювач, тому в якості нього була вибрана мікросхема LM321, що має в собі лише один операційний підсилювач, і завдяки цьому ще і менші розміри. Деякі характеристики LM321:

- Живлення: однополярне — 3-32В, двополярне — 1,5-16В;
- Смуга пропускання: 1МГц;
- Максимальна вихідна напруга: 26В, при однополярному живленні;
- Корпус: SOT5;

У даному пристрої операційний підсилювач використовується в якості диференційного підсилювача (Рис. 3.15).

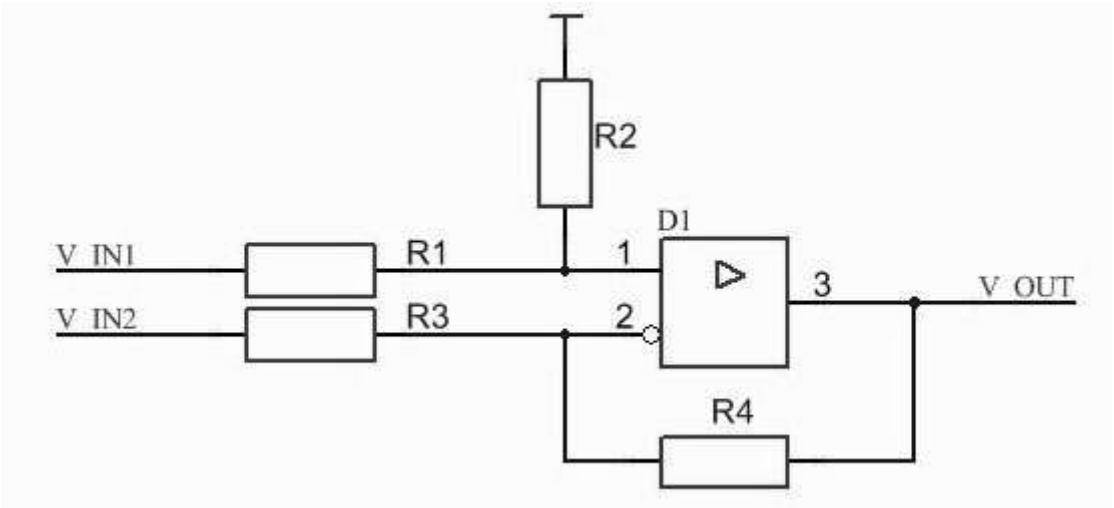


Рисунок 3.15 — Диференційний підсилювач на ОП

Номінали R_2 та R_4 — 1,5МОм, R_1 та R_3 — 1кОм, тоді у даній схемі виконується рівність $\frac{R_2}{R_1}=\frac{R_4}{R_3}$, тому вихідний сигнал можна порахувати за наступною формулою:

$$V_{OUT}=\left(V_{IN1}-V_{IN2}\right)\frac{R_2}{R_1} \text{ (3.1)},$$

де $\frac{R_2}{R_1}$ буде дорівнювати коефіцієнту підсилення.

Якщо підставити значення, то отримаємо коефіцієнт підсилення у 1500 разів.

Оскільки напруга на виході операційного підсилювача може сягати 9В, а на аналоговий вхід мікроконтролера не можна подавати більше за 3,3В, то потрібно цю напругу зменшити. Для цього використаємо дільник напруги та стабілітрон (Рис. 3.16).

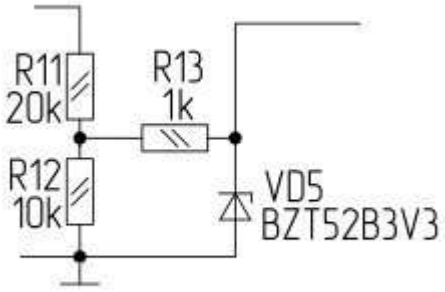


Рисунок 3.16 — Фрагмент схеми вихідного кола приймача

Дільник зменшить напругу до 3В, тобто на дві третини від 9В, а значить відношення резисторів $\frac{R_{11}}{R_{12}}$ у дільнику має дорівнювати $\frac{2}{1}$. Також потрібно враховувати потужність, яка буде розсіюватись на дільнику напруги. Ця потужність залежить від загального опору дільника, тому потрібно вибрати якомога більший опір, але такий, щоб АЦП мікроконтролера зміг зчитати сигнал.

Отже, для R_{11} використаємо резистор номіналом 20кОм, а для R_{12} — 10кОм. Тоді напруга на виході дільника буде дорівнювати:

$$V_{ВИХ} = \frac{V_{ВХ} R_{12}}{R_{11} + R_{12}} = \frac{9\,В \cdot 10\,кОм}{20\,кОм \cdot 10\,кОм} = 3\text{В} \quad (3.1)$$

Потужність, що розсіюється на дільнику:

$$I_Д = V_{ВХ}\left(R_{11} + R_{12}\right) = 9\,В \cdot \left(20\,кОм + 10\,кОм\right) = 300\,мкА \quad (3.2)$$

$$P_{РОЗ} = V_{ВХ} \cdot I_Д = 9\text{В} \cdot 300\text{мкА} = 2{,}7\text{мВт} \quad (3.3)$$

Використання стабілітрона з номінальною напругою стабілізації 3.3В гарантує, що значення напруги на вході АЦП не перевищить допустимий максимум. В якості такого стабілітрона використаємо BZT52B3V3, який має номінальну напругу стабілізації 3,3В та максимальний допустимий струм 118мА. Також перед стабілітроном необхідно встановити резистор, щоб запобігти перевищенню максимально допустимого струму. Цей резистор можна розрахувати за законом Ома:

$$R_{13} = \frac{V}{I} = \frac{9\text{В}}{10\text{мА}} = 900\,Ом \quad (3.4)$$

В результаті, отримуємо значення 900Ом, яке округлимо до 1КОм, тому що такі резистори вже використовуються в схемі.

Розроблена у цьому розділі схема представлена на рисунку 3.17.

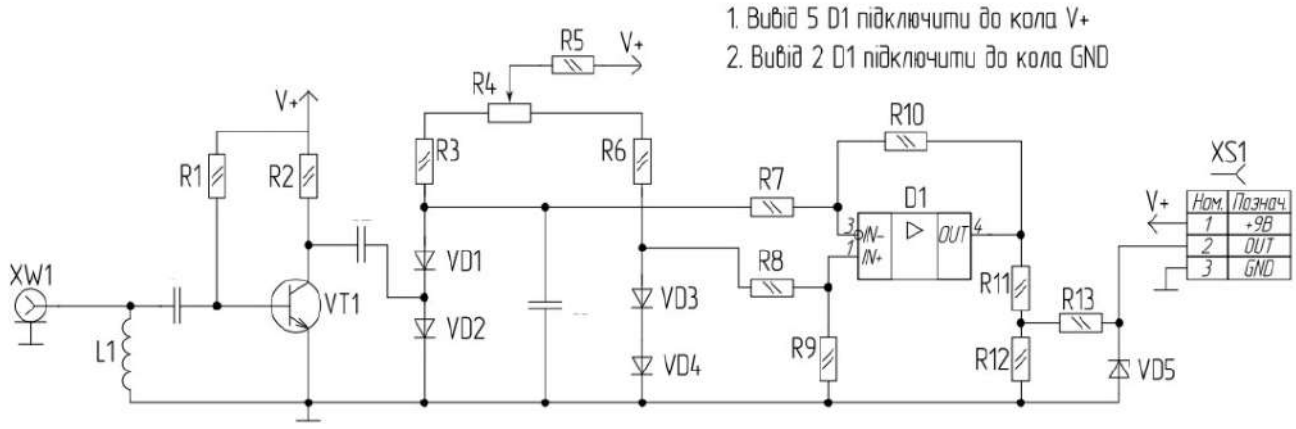


Рисунок 3.17 — Розроблена схема блоку приймача

РОЗДІЛ 4 РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Інструменти для розробки програмного забезпечення

В якості основного інструменту для розробки ПЗ для цього проекту використовується PlatformIO. PlatformIO — це сучасна екосистема для розробки програмного забезпечення вбудованих систем. Вона надає інструменти для керування проектами, компіляції, завантаження та налагодження програмного забезпечення для мікроконтролерів. PlatformIO підтримує більше ніж 1000 плат і 40 платформ, серед яких є Arduino, ESP8266, ESP32, STM32 та багато інших, що дозволяє розробляти програмне забезпечення для різних мікроконтролерів та процесорів без необхідності перемикатися між різними інструментами.

PlatformIO має вбудований менеджер бібліотек, що дозволяє легко знаходити, встановлювати та керувати бібліотеками, які необхідні для проекту. Більшість бібліотек підтримують одразу декілька плат/платформ, що полегшує перенесення проекту на інше апаратне забезпечення при необхідності.

Також великим плюсом є те, що PlatformIO інтегрується з різними середовищами розробки, включаючи Visual Studio Code, Atom, CLion та інші. Це дозволяє використовувати потужні редактори з підтримкою автодоповнення, налагодження та іншими зручностями, або ж навпаки, використовувати для розробки лише простий текстовий редактор.

PlatformIO є потужним інструментом для розробників вбудованих систем, забезпечуючи єдине середовище для всього циклу розробки програмного забезпечення: від написання коду до тестування та розгортання.

Libopencm3 — це вільна бібліотека з відкритим вихідним кодом для роботи з мікроконтролерами на базі ARM Cortex-M0, M0+, M3, M4 і M7. Вона призначена для спрощення розробки прошивок для мікроконтролерів різних

виробників, таких як STMicroelectronics (STM32), NXP (LPC), TI (Stellaris/Tiva), Atmel (SAM3/SAM4), та інших.

Бібліотека надає єдиний API для роботи з різними периферійними пристроями мікроконтролера, такими як GPIO, UART, SPI, I2C, ADC, таймери тощо. Це дозволяє переносити код між різними мікроконтролерами з мінімальними змінами. Крім того, бібліотека організована таким чином, що можливо підключати лише ті модулі, які дійсно потрібні, зменшуючи розмір кінцевого файлу прошивки.

Бібліотека сумісна з сучасними інструментами розробки, такими як компілятор GCC, налагоджувач GDB, система збірки програмного забезпечення Makefiles та інші.

Libopencm3 поширюється під ліцензією LGPL, що дозволяє використовувати її у комерційних та відкритих проектах з певними обмеженнями.

Libopencm3 є корисним інструментом для розробників, які працюють з різними мікроконтролерами на базі ARM Cortex-M, оскільки вона спрощує процес розробки, забезпечує високу портативність коду і знижує залежність від специфічного апаратного забезпечення виробника.

У цьому проєкті для керування завданнями, такими як зчитування даних з послідовного порту, підготовка та передача даних на дисплей, аналіз вхідних сигналів, буде використовуватись операційна система реального часу(RTOS).

RTOS, або Real-Time Operating System, є операційною системою, призначеною для управління апаратними ресурсами в режимі реального часу. Це означає, що вона здатна обробляти події або дані у визначені, передбачувані проміжки часу. RTOS зазвичай використовується в системах, де важлива швидка реакція та передбачуваність, таких як вбудовані системи, промислова автоматика, робототехніка та телекомунікації.

Основна характеристика RTOS полягає в її здатності забезпечувати гарантоване виконання завдань у суворо визначені терміни. Це досягається

завдяки ефективному управлінню пріоритетами завдань та перериваннями, що дозволяє RTOS швидко реагувати на зовнішні події. Крім того, RTOS часто має менший розмір і спрощену структуру в порівнянні з операційними системами загального призначення, що дозволяє їй працювати на апаратурі з обмеженими ресурсами[24].

У таких системах важливо, щоб критичні завдання виконувалися вчасно, інакше це може призвести до збоїв або небажаних наслідків. Наприклад, у системах управління автомобілями RTOS забезпечує своєчасну обробку сигналів від датчиків та відповідне керування двигуном і гальмами, що критично важливо для безпеки.

FreeRTOS (Free Real-Time Operating System) — це відкрита і безкоштовна операційна система реального часу (RTOS) для вбудованих систем. Вона є ядром для організації багатозадачності в системах з обмеженими ресурсами. FreeRTOS широко використовується в мікроконтролерах і інших вбудованих пристроях через свою легкість, гнучкість і низькі вимоги до апаратних ресурсів[25].

FreeRTOS спроектовано таким чином, щоб мінімізувати використання пам'яті. В основному коді операційної системи є ядро, яке займає дуже мало місця у флеш-пам'яті та оперативній пам'яті. Це робить FreeRTOS ідеальним для мікроконтролерів з обмеженими ресурсами, де економія пам'яті є критично важливою.

FreeRTOS дозволяє розробникам створювати кілька задач (потоків виконання), які можуть виконуватися паралельно. Це досягається за допомогою перемикання контексту, яке відбувається в момент, коли активна задача блокується або закінчує своє виконання, дозволяючи іншим задачам отримати доступ до процесорних ресурсів.

Також FreeRTOS забезпечує кілька механізмів для синхронізації та комунікації між задачами[26]:

- Семафори — використовуються для управління доступом до загальних ресурсів.

- М’ютекси — схожі на семафори, але забезпечують додаткові можливості, такі як пріоритетне наслідування.
- Черги повідомлень — дозволяють задачам обмінюватися даними через повідомлення.
- Події — використовуються для синхронізації задач на основі конкретних подій.

FreeRTOS надає 5 алгоритмів управління пам'яттю, що дозволяють динамічно виділяти і звільняти пам'ять:

- heap_1 — найпростіший, не дозволяє звільняти пам'ять.
- heap_2 — дозволяє звільнити пам'ять, але не об'єднує сусідні вільні блоки.
- heap_3 — просто робить стандартні функції malloc() і free() безпечними при використанні багатозадачності.
- heap_4 — об'єднує суміжні вільні блоки, щоб уникнути фрагментації.
- heap_5 — такий як heap_4, але з можливістю охоплювати кілька несуміжних областей пам'яті.

FreeRTOS може обробляти апаратні переривання, що дозволяє задачам реагувати на події в реальному часі. Апаратні переривання можуть викликати задачі або безпосередньо взаємодіяти з задачами через черги і семафори.

FreeRTOS підтримує широкий спектр апаратних платформ, включаючи ARM Cortex-M, AVR, PIC, MSP430, RISC-V і багато інших. Це забезпечує легкість перенесення коду між різними мікроконтролерами, що зменшує витрати на розробку та тестування.

Модульність системи FreeRTOS дозволяє розробникам додавати або змінювати її функціональність відповідно до потреб їхніх проектів. Наприклад, можна додати нові алгоритми планування задач або власні механізми синхронізації.

Не менш важливим є те, що FreeRTOS має велику та активну спільноту користувачів, які діляться своїм досвідом і допомагають один одному. Доступні різноманітні ресурси, такі як форуми, блоги, приклади коду, детальна документація, навчальні матеріали і книги. Офіційний вебсайт FreeRTOS також

надає безліч прикладів і посібників, що полегшує навчання і впровадження системи.

4.2 Розробка програмного забезпечення для блоку прийому та генерації сигналу

Початком кожної програми на мові програмування С є функція main, в ній спочатку потрібно налаштувати мікроконтролер на роботу з джерелом тактових сигналів частотою 8МГц, викликавши функцію бібліотеки libopencm3:

```
rcc_clock_setup_in_hse_8mhz_out_48mhz();
```

Далі налаштовуємо АЦП. Для цього вмикаємо сигнал тактування для порту А та налаштовуємо його виводи 1-4 як входи першого АЦП:

```
rcc_periph_clock_enable(RCC_GPIOA);
gpio_mode_setup(ADC1, GPIO_MODE_ANALOG, GPIO_PUPD_NONE, GPIO1 | GPIO2 | GPIO3 | GPIO4);
```

Далі вмикаємо сигнал тактування та розрядність першого АЦП:

```
rcc_periph_clock_enable(RCC_ADC1);
adc_set_clk_source(ADC1, ADC_CLKSOURCE_ADC);
adc_set_resolution(ADC1, ADC_RESOLUTION_12BIT);
adc_set_left_aligned(ADC1);
```

Вказуємо, які канали АЦП мають використовуватись, та встановлюємо час дискретизації:

```
uint8_t channels[] = {ADC_CHANNELS};
adc_set_regular_sequence(ADC1, sizeof(channels), channels);
adc_set_sample_time_on_all_channels(ADC1, ADC_SMPTIME_071DOT5);
adc_set_operation_mode(ADC1, ADC_MODE_SCAN);
adc_set_continuous_conversion_mode(ADC1);
```

Налаштовуємо прямий доступ до пам’яті для АЦП:

```
rcc_periph_clock_enable(RCC_DMA1);
dma_channel_reset(DMA1, DMA_CHANNEL1);
dma_set_peripheral_address(DMA1, DMA_CHANNEL1, (uint32_t) &ADC_DR(ADC1));
dma_set_memory_address(DMA1, DMA_CHANNEL1, (uint32_t) adc_dat);
dma_enable_memory_increment_mode(DMA1, DMA_CHANNEL1);
dma_set_peripheral_size(DMA1, DMA_CHANNEL1, DMA_CCR_PSIZE_16BIT);
dma_set_memory_size(DMA1, DMA_CHANNEL1, DMA_CCR_MSIZE_16BIT);
dma_set_priority(DMA1, DMA_CHANNEL1, DMA_CCR_PL_LOW);
dma_set_number_of_data(DMA1, DMA_CHANNEL1, sizeof(channels));
dma_enable_circular_mode(DMA1, DMA_CHANNEL1);
dma_enable_channel(DMA1, DMA_CHANNEL1);
adc_enable_dma(ADC1);
```

І нарешті, вмикаємо АЦП:

```
uint8_t i = 255;
while (i--);
adc_power_on(ADC1);
```

Наступним кроком, налаштовуємо таймер для генерації ШІМ сигналу:

```
rcc_periph_clock_enable(RCC_GPIOA);
gpio_mode_setup(PWM_PORT, GPIO_MODE_AF, GPIO_PUPD_NONE, PWM_PIN);
gpio_set_af(PWM_PORT, GPIO_AF1, PWM_PIN);

rcc_periph_clock_enable(PWM_RCC);
```

```
rcc_periph_reset_pulse(RST_TIM3);
timer_set_mode(PWM_TIM, TIM_CR1_CKD_CK_INT, TIM_CR1_CMS_EDGE, TIM_CR1_DIR_UP);
timer_set_prescaler(PWM_TIM, rcc_apb1_frequency);
timer_disable_preload(PWM_TIM);
timer_continuous_mode(PWM_TIM);
timer_set_period(PWM_TIM, TIM_PSC_DIV);

timer_set_oc_mode(PWM_TIM, TIM_OC1, TIM_OCM_PWM2);
timer_enable_oc_output(PWM_TIM, TIM_OC1);
timer_set_oc_value(PWM_TIM, TIM_OC1, TIM_PSC_DIV);

timer_enable_break_main_output(PWM_TIM);
```

Налаштування UART. Вмикаємо сигнал тактування для порту A та першого UART і налаштовуємо виводи 9 і 10 на роботу з UART:

```
rcc_periph_clock_enable(RCC_GPIOA);
rcc_periph_clock_enable(RCC_USART1);
gpio_mode_setup(GPIOA, GPIO_MODE_AF, GPIO_PUPD_NONE, GPIO9 | GPIO10);
gpio_set_af(GPIOA, GPIO_AF1, GPIO9 | GPIO10);
```

Встановлюємо параметри UART:

```
usart_set_baudrate(U_PORT, 115200);
usart_set_databits(U_PORT, 8);
usart_set_parity(U_PORT, USART_PARITY_NONE);
usart_set_stopbits(U_PORT, USART_CR2_STOPBITS_1);
usart_set_mode(U_PORT, USART_MODE_TX_RX);
usart_set_flow_control(U_PORT, USART_FLOWCONTROL_NONE);
```

Вмикаємо переривання від UART:

```
nvic_enable_irq(NVIC_USART1_IRQ);
usart_enable_rx_interrupt(U_PORT);
```

Вмикаємо UART:

```
usart_enable(U_PORT);
```

Створюємо завдання для зчитування даних по UART, аналізу значень з АЦП, генерації ШІМ з потрібними параметрами та запускаємо планувальник завдань:

```
if (xTaskCreate(&usart_task, "u", 127, NULL, 8, NULL) != pdPASS) {
   configASSERT(0);
}
if (xTaskCreate(&adc_analyze_task, "a", 127, NULL, 8, NULL) != pdPASS) {
   configASSERT(0);
}
if (xTaskCreate(&saw_task, "s", 127, NULL, 8, &saw_task_hdl) != pdPASS) {
   configASSERT(0);
}
vTaskSuspend(saw_task_hdl);
vTaskStartScheduler();
```

Тепер розглянемо кожне завдання окремо.

У завданні зчитування даних з блоку керування по UART спочатку зчитуємо перший символ, і далі перевіряємо, чи є він символом «#», з якого починається повідомлення. Якщо початок повідомлення знайдений, то зчитуємо решту повідомлення у буфер. Це повідомлення складається із двох значень: перше вказує, чи була генерація була запущена вручну, а друге — контрольна сума для перевірки цілісності даних. Якщо перевірка контрольної суми показала, що повідомлення ціле, то відповідно до першого значення, запускаємо чи

зупиняємо генерацію шуму. Також, в кінці додаємо затримку, під час якої планувальник FreeRTOS поставить на виконання інші завдання. Ці всі дії необхідно виконувати циклічно поки працює пристрій.

Код завдання зчитування даних по UART:

```
char ch = usart_getc(false);
if (ch == '#') {
    char buf[5];
    char *ptr = buf;

    *(ptr++) = usart_getc(1);
    *(ptr++) = ' ';
    *(ptr++) = usart_getc(1);
    *(ptr++) = usart_getc(1);
    *(ptr++) = 0;

    char *end;
    cdp.force_gen = strtol(buf, &end, 10);
    cdp.hash = strtol(end, &end, 10);

    if (cdp_check_hash(&cdp)) {
        rdp.force_gen = cdp.force_gen;
        if (cdp.force_gen) {
            timer_enable_counter(PWM_TIM);
            vTaskResume(saw_task_hdl);
        } else {
            vTaskSuspend(saw_task_hdl);
            timer_set_oc_value(PWM_TIM, TIM_OC1, TIM_PSC_DIV);
            timer_disable_counter(PWM_TIM);
        }
    }
```

```
    }
    vTaskDelay(20 / portTICK_PERIOD_MS);
```

У завданні аналізу значень з АЦП спочатку запускаємо процес зчитування (дискретизації) значення напруги на входах АЦП та чекаємо поки він завершиться. Далі порівнюємо значення отримані з АЦП з пороговим значенням. Якщо хоча б одне значення є більшим за порогове, то запускаємо генерацію, а якщо ні — зупиняємо. Також надсилаємо зчитані значення до блоку керування через UART. Ці всі дії необхідно виконувати циклічно поки працює пристрій.

Код завдання аналізу значень з АЦП:

```
    dma_clear_interrupt_flags(DMA1, DMA_CHANNEL1, DMA_TCIF | DMA_HTIF | DMA_TEIF);
    adc_start_conversion_regular(ADC1);
    while (!(dma_get_interrupt_flag(DMA1, DMA_CHANNEL1, DMA_TCIF) ||
           dma_get_interrupt_flag(DMA1, DMA_CHANNEL1, DMA_TEIF)));

    if (dma_get_interrupt_flag(DMA1, DMA_CHANNEL1, DMA_TEIF)) {
      vTaskDelay(100 / portTICK_PERIOD_MS);
    }

    if (adc_dat[0] * 100 / 65536 > THRESHOLD || adc_dat[1] * 100 / 65536 > THRESHOLD ||
        adc_dat[2] * 100 / 65536 > THRESHOLD || adc_dat[3] * 100 / 65536 > THRESHOLD) {
      rdp.gen_on = true;
      if (!rdp.force_gen) {
        timer_enable_counter(PWM_TIM);
        vTaskResume(saw_task_hdl);
      }
    } else {
```

```
        rdp.gen_on = false;
        if (!rdp.force_gen) {
            vTaskSuspend(saw_task_hdl);
            timer_set_oc_value(PWM_TIM, TIM_OC1, TIM_PSC_DIV);
            timer_disable_counter(PWM_TIM);
        }
    }

    memcpy(rdp.channels, adc_dat, sizeof(uint16_t) * 4);
    rdp_gen_hash(&rdp);
    send_packet();

    vTaskDelay(50 / portTICK_PERIOD_MS);
```

У завданні, що керує генерацією ШІМ сигналу, поступово збільшуємо коефіцієнт заповнення до максимального значення:

```
    while (1) {
        for (int i = 0; i < TIM_PSC_DIV; ++i)
            timer_set_oc_value(PWM_TIM, TIM_OC1, i);

        vTaskDelay(1);
    }
```

4.3 Розробка програмного забезпечення для блоку керування

Спочатку налаштовуємо тактування від внутрішнього джерела тактових сигналів:

```
    rcc_clock_setup_in_hsi_out_48mhz();
```

Щоб працювати з кнопкою ручного пуску, необхідно увімкнути тактування для порту A, та налаштувати вивід мікроконтролера, до якого вона буле підключена, як вхід з підтяжкою до напруги живлення (3.3В). Це робиться наступним чином:

```
rcc_periph_clock_enable(RCC_GPIOA);
gpio_mode_setup(GPIOA, GPIO_MODE_INPUT, GPIO_PUPD_PULLUP, GPIO0);
```

Налаштування UART. Спочатку вмикаємо тактування для порту A та для першого UART, далі налаштовуємо виводи 2 і 3 для роботи з UART:

```
rcc_periph_clock_enable(RCC_GPIOA);
rcc_periph_clock_enable(RCC_USART1);
gpio_mode_setup(GPIOA, GPIO_MODE_AF, GPIO_PUPD_NONE, GPIO2 | GPIO3);
gpio_set_af(GPIOA, GPIO_AF1, GPIO2 | GPIO3);
```

Встановлюємо параметри UART, такі як швидкість передачі, режим роботи, паритет, так інші:

```
usart_set_baudrate(U_PORT, 115200);
usart_set_databits(U_PORT, 8);
usart_set_parity(U_PORT, USART_PARITY_NONE);
usart_set_stopbits(U_PORT, USART_CR2_STOPBITS_1);
usart_set_mode(U_PORT, USART_MODE_TX_RX);
usart_set_flow_control(U_PORT, USART_FLOWCONTROL_NONE);
```

Вмикаємо переривання, яке сповістить про те, що є нові дані які можна прийняти по UART:

```
nvic_enable_irq(NVIC_USART1_IRQ);
usart_enable_rx_interrupt(U_PORT);
```

Вмикаємо UART після встановлення налаштувань:

```
usart_enable(U_PORT);
```

Налаштування передачі даних по шині I^2C, котра буде використовуватись для комунікації з дисплеєм. Спочатку вмикаємо тактування для драйвера I^2C та подаємо на нього сигнал скидання:

```
rcc_periph_clock_enable(RCC_I2C1);
rcc_periph_clock_enable(RCC_GPIOA);
rcc_set_i2c_clock_hsi(I2C1);
rcc_periph_reset_pulse(RST_I2C1);
```

Далі налаштовуємо виводи 9 і 10 порту A для роботи з I^2C:

```
gpio_mode_setup(GPIOA, GPIO_MODE_AF, GPIO_PUPD_PULLUP, GPIO9 |
GPIO10);
gpio_set_af(GPIOA, GPIO_AF4, GPIO9 | GPIO10);
```

Встановлюємо швидкість передачі та вмикаємо фільтри шуму:

```
i2c_enable_analog_filter(I2C1);
i2c_set_digital_filter(I2C1, 0);
i2c_set_speed(I2C1, i2c_speed_sm_100k, 8);
i2c_set_7bit_addr_mode(I2C1);
```

Вмикаємо драйвер I2C після встановлення його налаштувань:

```
i2c_peripheral_enable(I2C1);
```

Після того як всі периферійні пристрої налаштовано, створюємо задачі для зчитування стану кнопки, відображення даних на дисплей, передачі по UART та запускаємо планувальник:

```
if (xTaskCreate(&btn_task, "b", 127, NULL, 7, NULL) != pdPASS) {
    configASSERT(0);
```

```
}
if (xTaskCreate(&lcd_task, "l", 127, NULL, 7, NULL) != pdPASS) {
   configASSERT(0);
}
if (xTaskCreate(&usart_task, "u", 127, NULL, 6, NULL) != pdPASS) {
   configASSERT(0);
}

vTaskStartScheduler();
```

Тепер розглянемо кожне завдання окремо.

У завданні зчитування стану кнопки, періодично викликаємо функцію btn_tick, код якої наведено нижче. В цій функції перевіряється стан кнопки, та протягом якого часу вона була у натисненому стані. Якщо кнопка була у натисненому стані менше ніж 70мс, то це можна вважати шумом, а якщо більше, то вважаємо, що кнопка була натиснена оператором пристрою, та виконуємо дію: запускаємо генерацію шуму, якщо вона ще не запущена, або навпаки — вимикаємо, якщо генерація вже запущена. Нижче наведено код функції btn_tick, та кілька змінних, що необхідні для її роботи:

```
bool been_pressed = false;
uint32_t press_time = 0;
bool click_started = false;

static void btn_tick(void) {
   if (!been_pressed && !gpio_get(GPIOA, GPIO0)) {
      press_time = millis();
      been_pressed = true;
      return;
   } else if (been_pressed) {
      if ((millis() - press_time) >= 70) {
```

```
            if (!gpio_get(GPIOA, GPIO0)) {
                if (!click_started) {
                    click_started = true;
                    do_click_action();
                }
            } else {
                been_pressed = false;
                click_started = false;
            }
        }
    }
}
```

У завданні відображення даних на дисплеї спочатку ініціалізуємо дисплей:

```
lcd_init();
```

Далі, з періодичністю 200мс, оновлюємо текст на дисплеї. Якщо дані від блоку прийому сигналу та генерації шуму прийняті без помилок, то відображаємо їх, а якщо при передачі даних виникли помилки, то виводимо на дисплей помилку. Код оновлення тексту на дисплеї наведено нижче:

```
while (1) {
    lcd_clear();

    char buf[17];
    if (rdp_valid) {
            snprintf(buf, 17, "N:%hu S:%hu", rdp.channels[0] * 100 / 65535, rdp.channels[1] * 100 / 65535);
        lcd_put_cur(0, 0);
        lcd_send_string(buf);
```

```
                snprintf(buf, 17, "W:%hu E:%hu", rdp.channels[2] * 100 / 65535,
rdp.channels[3] * 100 / 65535);
            lcd_put_cur(1, 0);
            lcd_send_string(buf);

            if (rdp.force_gen) {
               lcd_put_cur(1, 14);
               lcd_send_string("f");
            }
            if (rdp.gen_on) {
               lcd_put_cur(1, 15);
               lcd_send_string("g");
            }
         } else {
            snprintf(buf, 17, "%s", "Invalid packet");
            lcd_put_cur(0, 0);
            lcd_send_string(buf);
         }

         vTaskDelay(200 / portTICK_PERIOD_MS);
      }
```

Приклад зображення, яке отримаємо на дисплеї наведено на рисунку 3. Буквами N, S, W та E позначено сторони світу північ, південь, захід та схід відповідно.

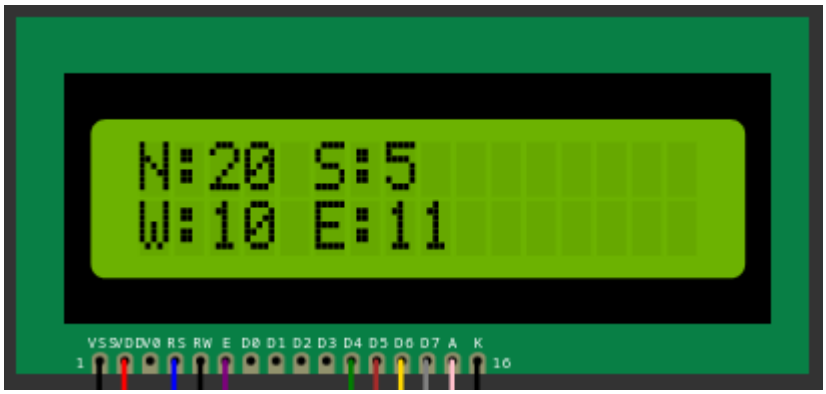


Рисунок 4.1 — Приклад зображення на дисплеї

Код функцій, що використовуються для комунікації з дисплеєм:

```
void lcd_send_cmd(char cmd) {
    char data_u, data_l;
    uint8_t data_t[4];
    data_u = (cmd & 0xf0);
    data_l = ((cmd << 4) & 0xf0);
    data_t[0] = data_u | 0x0C;  //en=1, rs=0
    data_t[1] = data_u | 0x08;  //en=0, rs=0
    data_t[2] = data_l | 0x0C;  //en=1, rs=0
    data_t[3] = data_l | 0x08;  //en=0, rs=0
    i2c_transfer7(I2C1, I2C_LCD_ADDR, data_t, sizeof(data_t), NULL, 0);
}

void lcd_send_data(char data) {
    char data_u, data_l;
    uint8_t data_t[4];
    data_u = (data & 0xf0);
    data_l = ((data << 4) & 0xf0);
    data_t[0] = data_u | 0x0D;  //en=1, rs=0
    data_t[1] = data_u | 0x09;  //en=0, rs=0
    data_t[2] = data_l | 0x0D;  //en=1, rs=0
```

```
    data_t[3] = data_l | 0x09;  //en=0, rs=0
    i2c_transfer7(I2C1, I2C_LCD_ADDR, data_t, sizeof(data_t), NULL, 0);
}

void lcd_clear(void) {
    lcd_send_cmd(0x80);
    for (int i = 0; i < 70; i++) {
        lcd_send_data(' ');
    }
}

void lcd_put_cur(int row, int col) {
    switch (row) {
        case 0:
            col |= 0x80;
            break;
        case 1:
            col |= 0xC0;
            break;
    }

    lcd_send_cmd(col);
}
```

Код функція, ініціалізації дисплею у 4-бітному режимі:

```
void lcd_init(void) {
    delay(50);
    lcd_send_cmd(0x30);
    delay(5);
    lcd_send_cmd(0x30);
```

```
    delay(1);
    lcd_send_cmd(0x30);
    delay(10);
    lcd_send_cmd(0x20);
    delay(10);

    lcd_send_cmd(0x28);
    delay(1);
    lcd_send_cmd(0x08);
    delay(1);
    lcd_send_cmd(0x01);
    delay(1);
    delay(1);
    lcd_send_cmd(0x06);
    delay(1);
    lcd_send_cmd(0x0C);
}
```

Допоміжна функція для надсилання строки:

```
void lcd_send_string(char *str) {
    while (*str) lcd_send_data(*str++);
}
```

У завданні зчитування даних від блоку прийому та генерації шуму по UART зчитуємо перший символ, і далі перевіряємо, чи є він символом «#», з якого має починатися повідомлення. Якщо початок повідомлення знайдений, то зчитуємо решту повідомлення у буфер. Це повідомлення складається із чотирьох значень, які є рівнями сигналу, двох значень що вказують, чи запущена генерація, і спосіб запуску: автоматично чи вручну. І останнім значенням в повідомлені є його контрольна сума. Якщо перевірка контрольної суми показала, що

повідомлення ціле, то зберігаємо його, щоб відобразити на дисплеї. Також, в кінці додаємо затримку, під час якої планувальник FreeRTOS поставить на виконання інші завдання. Код завдання наведено нижче:

```
static void usart_task(void *args) {
    while (1) {
        char ch = usart_getc(false);
        if (ch == '#') {
            char buf[50];
            char *ptr = buf;
            uint8_t i;
            if_begin:

            for (i = 0; i < 5; i++) {
                *ptr = usart_getc(1);
                if (*ptr == '#') goto if_begin;
                ptr++;
            }
            *(ptr++) = ' ';
            for (i = 0; i < 5; i++) {
                *ptr = usart_getc(1);
                if (*ptr == '#') goto if_begin;
                ptr++;
            }
            *(ptr++) = ' ';
            for (i = 0; i < 5; i++) {
                *ptr = usart_getc(1);
                if (*ptr == '#') goto if_begin;
                ptr++;
            }
            *(ptr++) = ' ';
```

```
for (i = 0; i < 5; i++) {
   *ptr = usart_getc(1);
   if (*ptr == '#') goto if_begin;
   ptr++;
}
*(ptr++) = ' ';
*ptr = usart_getc(1);
if (*ptr == '#') goto if_begin;
ptr++;
*(ptr++) = ' ';
*ptr = usart_getc(1);
if (*ptr == '#') goto if_begin;
ptr++;
*(ptr++) = ' ';
*ptr = usart_getc(1);
if (*ptr == '#') goto if_begin;
ptr++;
*ptr = usart_getc(1);
if (*ptr == '#') goto if_begin;
ptr++;
*(ptr++) = 0;

char *end;
rdp.ch1 = strtol(buf, &end, 10);
rdp.ch2 = strtol(end, &end, 10);
rdp.ch3 = strtol(end, &end, 10);
rdp.ch4 = strtol(end, &end, 10);
rdp.gen_on = strtol(end, &end, 10);
rdp.force_gen = strtol(end, &end, 10);
rdp.hash = strtol(end, &end, 10);
```

```
            rdp_valid = rdp_check_hash(&rdp);
        }
        vTaskDelay(20 / portTICK_PERIOD_MS);
    }
}
```

При розробці програмного забезпечення було складено алгоритм роботи зображений на рисунку 4.2.

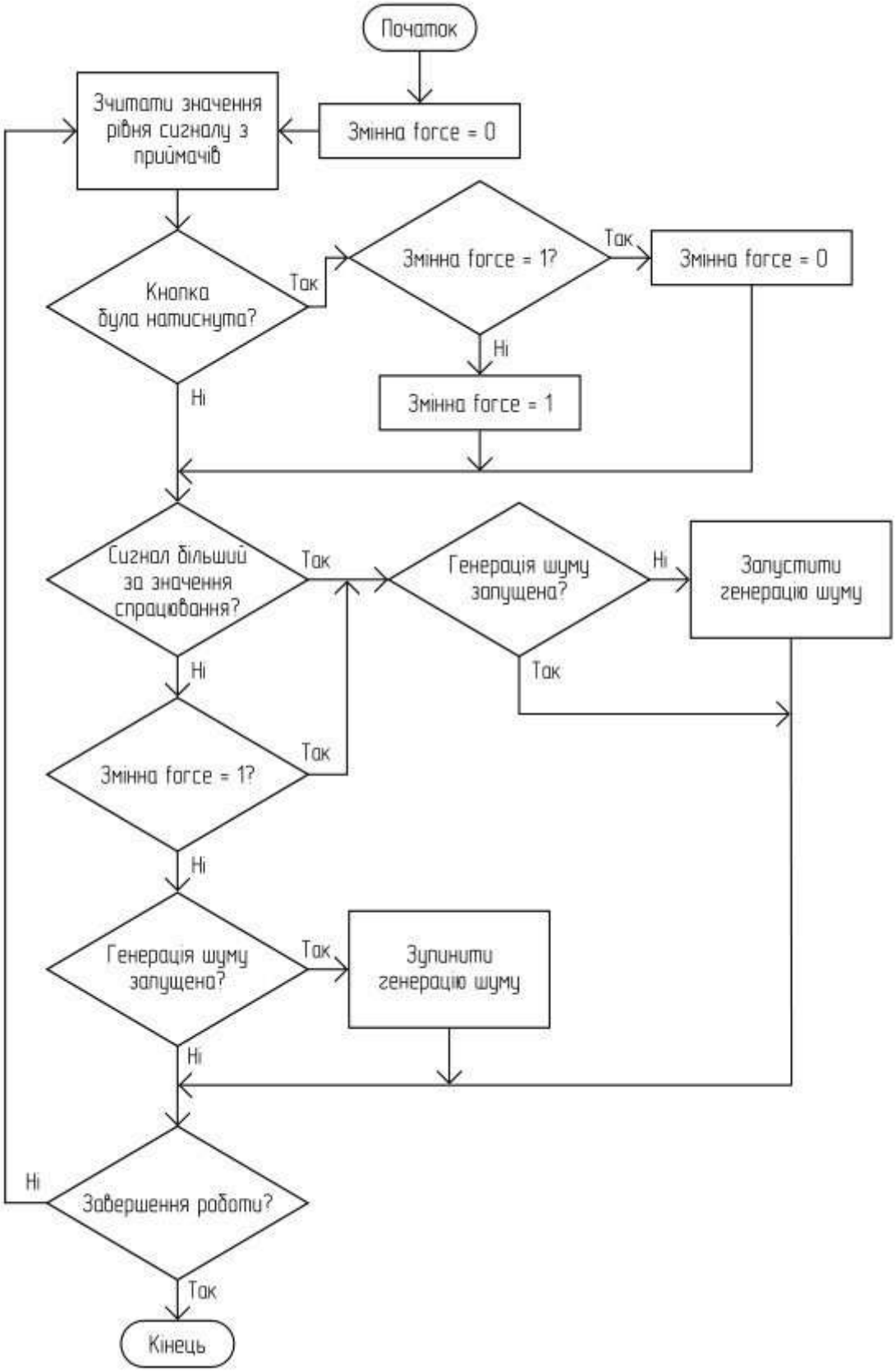


Рисунок 4.2 — Алгоритм роботи розробленого пристрою

ВИСНОВКИ

В ході дипломного проектування було розроблено засіб автоматизованого захисту від несанкціонованого спостереження.

Відповідно до мети, під час проектування розроблено креслення структурної схеми, схеми електричної принципової та алгоритму роботи. Також було розроблено програмне забезпечення для мікроконтролерів, що використовуються в проекті.

У першому розділі розглянуто можливі способи вирішення проблеми несанкціонованого відеоспостереження, такі як радіотехнічне, візуальне та акустичне виявлення, а також розглядаються сфери використання пристрою.

У другому розділі, відповідно до поставлених вимог, розроблено схему електричну структурну всього пристрою, а також обрано апаратну платформу та периферійні пристрої.

В третьому розділі обрано елементи схеми електричної принципової та виконано розрахунки електричних кіл, блоку керування, блоку прийому сигналу та генерації шуму, блоку приймача.

В четвертому розділі розроблено алгоритм роботи пристрою та програмне забезпечення для мікроконтролерів, з використанням операційної системи реального часу FreeRTOS.

ПЕРЕЛІК ПОСИЛАНЬ

1. Стаття про системи виявлення дронів [Електронний ресурс]. — Режим доступу: https://www.bezpeka-shop.com/ua/blog/obzor/sistemy-obnaruzheniya-dronov-i-protivodronnye-sistemy/
2. Everything You Need To Know About GPS L1, L2, and L5 Frequencies [Електронний ресурс]. — Режим доступу: https://gisresources.com/everything-you-need-to-know-about-gps-l1-l2-and-l5-frequencies/
3. Опис можливостей системи DroneTracker [Електронний ресурс]. — Режим доступу: https://www.dedrone.com/products/drone-detection-software
4. Drone AI Technology: How It Works; Why It Matters [Електронний ресурс]. — Режим доступу: https://consortiq.com/uas-resources/drone-ai-technology-how-it-works-why-it-matters
5. М. В. Криховецький Методи виявлення дронів на базі нейронних мереж: стаття, Прикарпатський національний університет імені Василя Стефаника, Україна, 2023.
6. Seongjoon Park, Hyeong Tae Kim, Sangmin Lee, Hyeontae Joo, Hwangnam Kim, Survey on Anti-Drone Systems: Components, Designs, and Challenges, Department of Electrical Engineering, Korea University, Seoul 02841, South Korea, 2021.
7. Системи виявлення та захисту від безпілотників: огляд і програмно-конфігуроване рішення на основі радіо: стаття, Бухарестський політехнічний університет, 2022.
8. Документація на мікроконтролер STM32F030K6 [Електронний ресурс]. — Режим доступу: https://www.st.com/resource/en/datasheet/stm32f030f4.pdf
9. Документація на мікросхему PCF8574T [Електронний ресурс]. — Режим доступу: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf

10. Документація на мікросхему MAX3485 [Електронний ресурс]. — Режим доступу: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX3483-MAX3491.pdf
11. Вказівки щодо правильного підключення RS-485 [Електронний ресурс]. — Режим доступу: https://pdfserv.maximintegrated.com/en/an/Guidelines_Proper_Wiring_Rs485_Network.pdf
12. Документація на лінійні регулятори напруги серії AMS1117 [Електронний ресурс]. — Режим доступу: http://www.advanced-monolithic.com/pdf/ds1117.pdf
13. Документація на лінійні регулятори напруги серії L78 [Електронний ресурс]. — Режим доступу:https://www.st.com/resource/en/datasheet/l78.pdf
14. Документація на ГКН YSGM081008 [Електронний ресурс]. — Режим доступу:https://www.javanelec.com/CustomAjax/GetAppDocument/2870fda8-b2e9-4451-b143-03ce10b37618
15. Документація на транзистор BFR92P [Електронний ресурс]. — Режим доступу: https://www.infineon.com/dgdl/Infineon-BFR92P-DS-v01_01-en.pdf?fileId=db3a30431400ef68011426fd178506a6
16. Документація на діод 1N4148W [Електронний ресурс]. — Режим доступу: https://www.vishay.com/docs/86356/1n4148w.pdf
17. Документація на мікросхему LM321 [Електронний ресурс]. — Режим доступу: https://www.ti.com/lit/ds/symlink/lm321.pdf
18. Схеми підключення операційних підсилювачів [Електронний ресурс]. — Режим доступу: https://habr.com/articles/508530/
19. Документація на стабілітрон BZT52 [Електронний ресурс]. — Режим доступу: https://www.vishay.com/docs/86342/bzt52_series.pdf
20. Домашня сторінка проекту libopencm3 [Електронний ресурс]. — Режим доступу: https://github.com/libopencm3/libopencm3/wiki
21. Приклади програм з використанням libopencm3 [Електронний ресурс]. — Режим доступу: https://github.com/libopencm3/libopencm3-examples

22. Getting started with PlatformIO - Better than ArduinoIDE [Електронний ресурс]. — Режим доступу: https://dronebotworkshop.com/platformio/
23. Що таке PlatformIO? [Електронний ресурс]. — Режим доступу: https://docs.platformio.org/en/latest/what-is-platformio.html
24. What is a real-time operating system (RTOS) [Електронний ресурс]. — Режим доступу: https://www.techtarget.com/searchdatacenter/definition/real-time-operating-system
25. What is FreeRTOS? [Електронний ресурс]. — Режим доступу: https://docs.aws.amazon.com/freertos/latest/userguide/what-is-freertos.html
26. Документація на API FreeRTOS [Електронний ресурс]. — Режим доступу: https://www.freertos.org/a00106.html

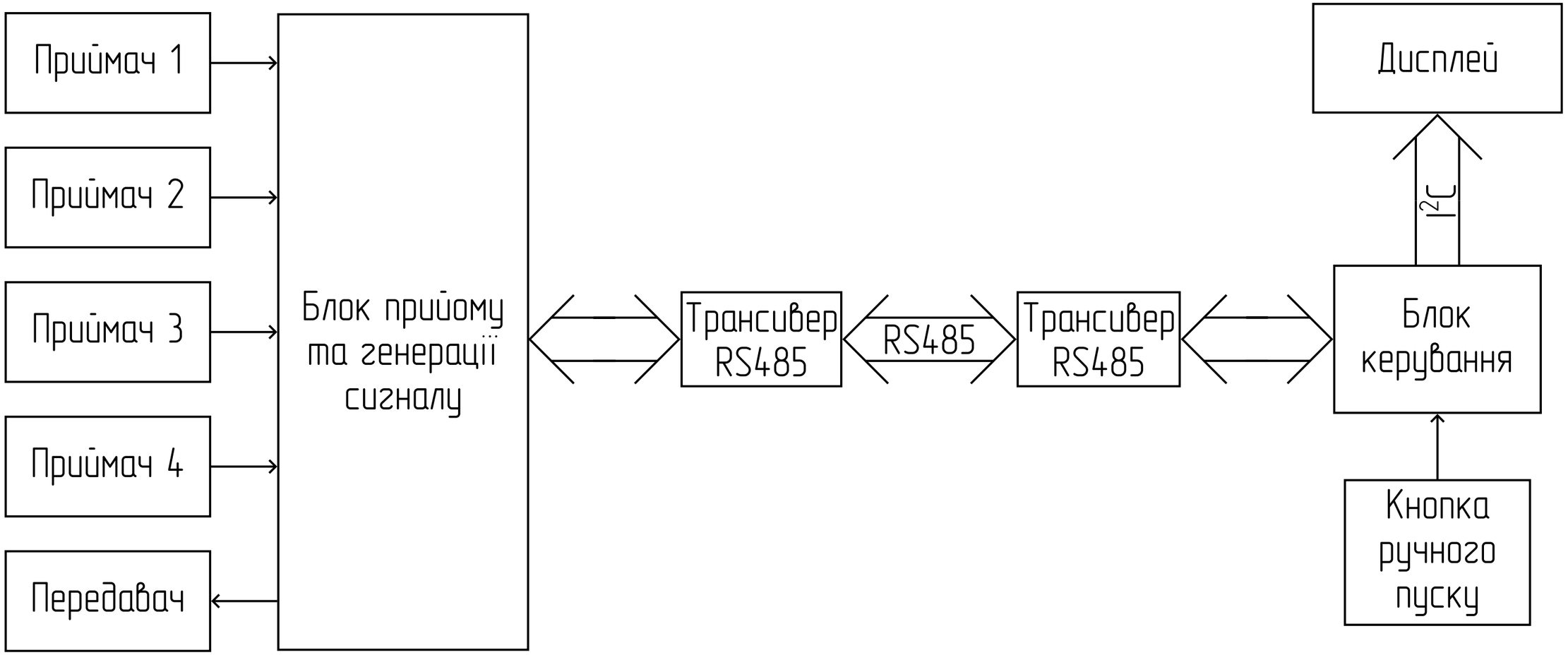
Поз. Познач.	Найменування	Кіл.	Примітка
	Блок приймача	4	
	Конденсатори		
C1,C2	4,7нФ ±5% 0805	2	
C3	1мкФ ±5% 0805	1	
D2	Мікросхема LM321	1	
L1	Котушка індуктивності 470нГ ±5% 0805	1	
	Резистори		
R1	100кОм ±5% 0805	1	
R2	1,2кОм ±5% 0805	1	
R3	240кОм ±5% 0805	1	
R4	10кОм 3362	1	Підстроювальний
R5	47кОм ±5% 0805	1	
R6	240кОм ±5% 0805	1	
R7,R8	1кОм ±5% 0805	2	
R9,R10	1,5МОм ±5% 0805	2	
R11	20кОм ±5% 0805	1	
R12,R13	10кОм ±5% 0805	2	
VD1-VD4	Діод 1N4148W	4	
VD5	Стабілітрон BZT52B3V3	1	
VT1	Транзистор BFR92P	1	

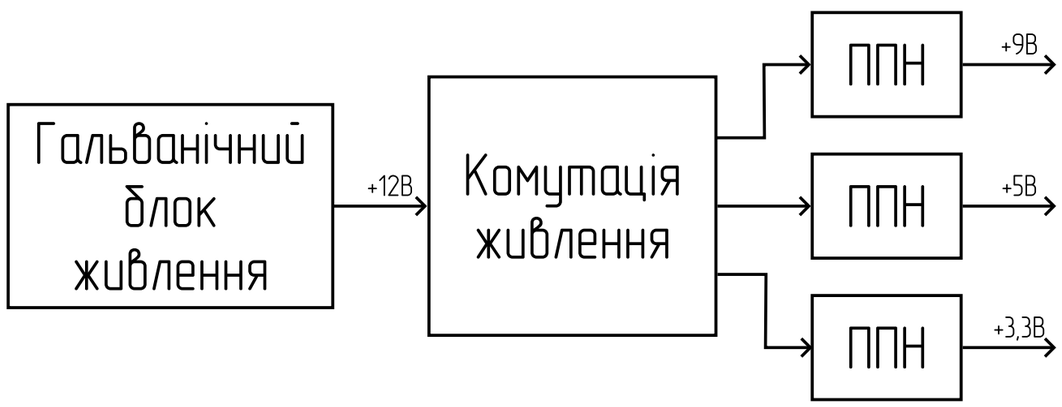
Зм	Лист	№ докум.	Підп.	Дата	КПТР.020051.01.03.ПЕ3		
Розробив		Разовий О.О.			Інформаційні технології проектування телекомунікаційних пристроїв	Літ.	Аркуш 1 / Аркушів 3
Перевірив		Петрушак В.С.				н	
Н.контр.		Пивовар О. С.			Перелік елементів	ХНУ, ФІТ	
Затверд.		Підченко С.К.					

Поз. Познач.	Найменування	Кіл	Примітка
XS1	Роз’єм JST PH-3	1	
XW1	Роз’єм KH-SMA-K513-G		
	Блок прийому сигналу та генерації шуму	1	
	Конденсатори		
C1	100мкФ ±10% 16В	1	
C2	22мкФ ±10% 16В	1	
C3	10мкФ ±10%16В	1	
C4	100нФ ±10% 0805	1	
C5	22мкФ ±10%16В	1	
C6	100мкФ ±10%	1	
C7, C8	22пФ ±1% 0805	2	
	Мікросхеми		
D1	AMS1117-5.0	1	
D2	MAX3485	1	
D3	L7809	1	
D4	AMS1117-3.3	1	
D5	STM32F030K6T6	1	
D6	YSGM081008	1	
R1	Резистор 120Ом ±2% 0805	1	
R2	Резистор 10кОм ±5% 0805	1	
	Роз’єми		
XS1	JST PH-6	1	
XS2	*WJ124-3.81-3P*	1	
XS3	WJ500V-5.08-2P-14-00A	1	

Поз. Познач.	Найменування	Кіл	Примітка
XS4	B-2100S04P-A110	1	
XW5	KH-SMA-K513-G	1	
ZQ1	Кварцовий резонатор 8МГц HC-49S-SMD	1	
	Блок керування	1	
	Конденсатори		
C1	22пФ ±1% 0805	1	
C2	27мкФ ±10%16В	1	
C3	22пФ ±1% 0805	1	
C4,C5	22мкФ ±10%16В	2	
	Мікросхеми		
D1	AMS1117-5.0	1	
D2	STM32F030K6T6	1	
D3	AMS1117-3.3	1	
D4	MAX3485	1	
R1	Резистор 10кОм ±5% 0805	1	
R2	Резистор 120Ом ±2% 0805	1	
	Роз'єми		
XS1	WJ500V-5.08-2P-14-00A	1	
XS2,XS3	B-2100S04P-A110	2	
XS4	*WJ124-3.81-3P*	1	
ZQ1	Кварцовий резонатор 8МГц HC-49S-SMD	1	

КПТР.020051.01.03.Е1

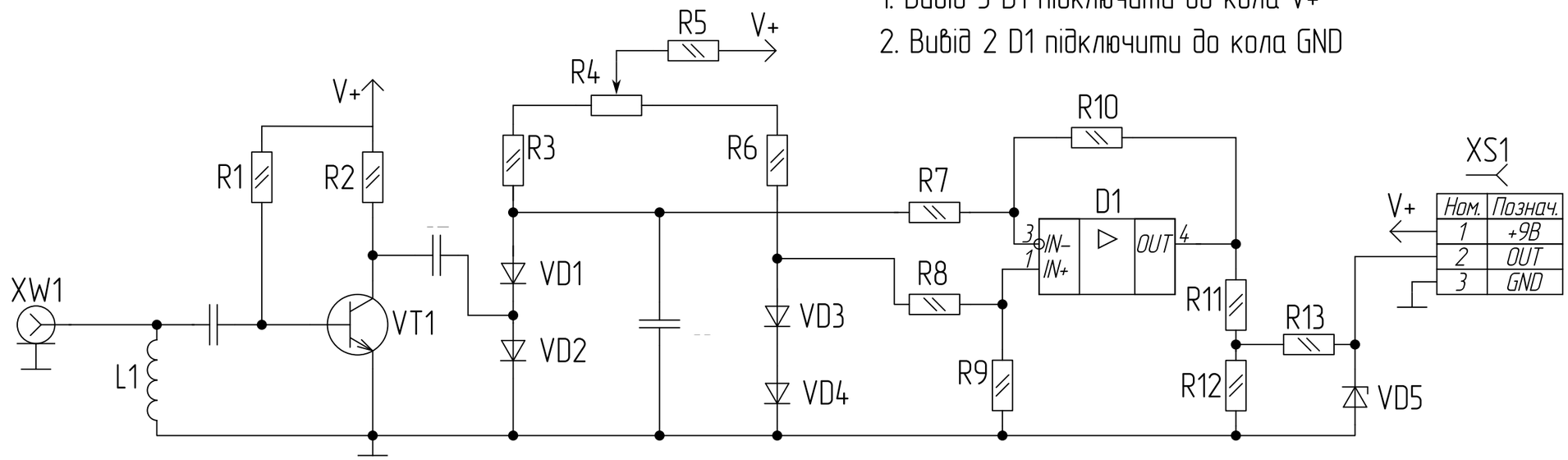




ППН – Понижаючий перетворювач напруги

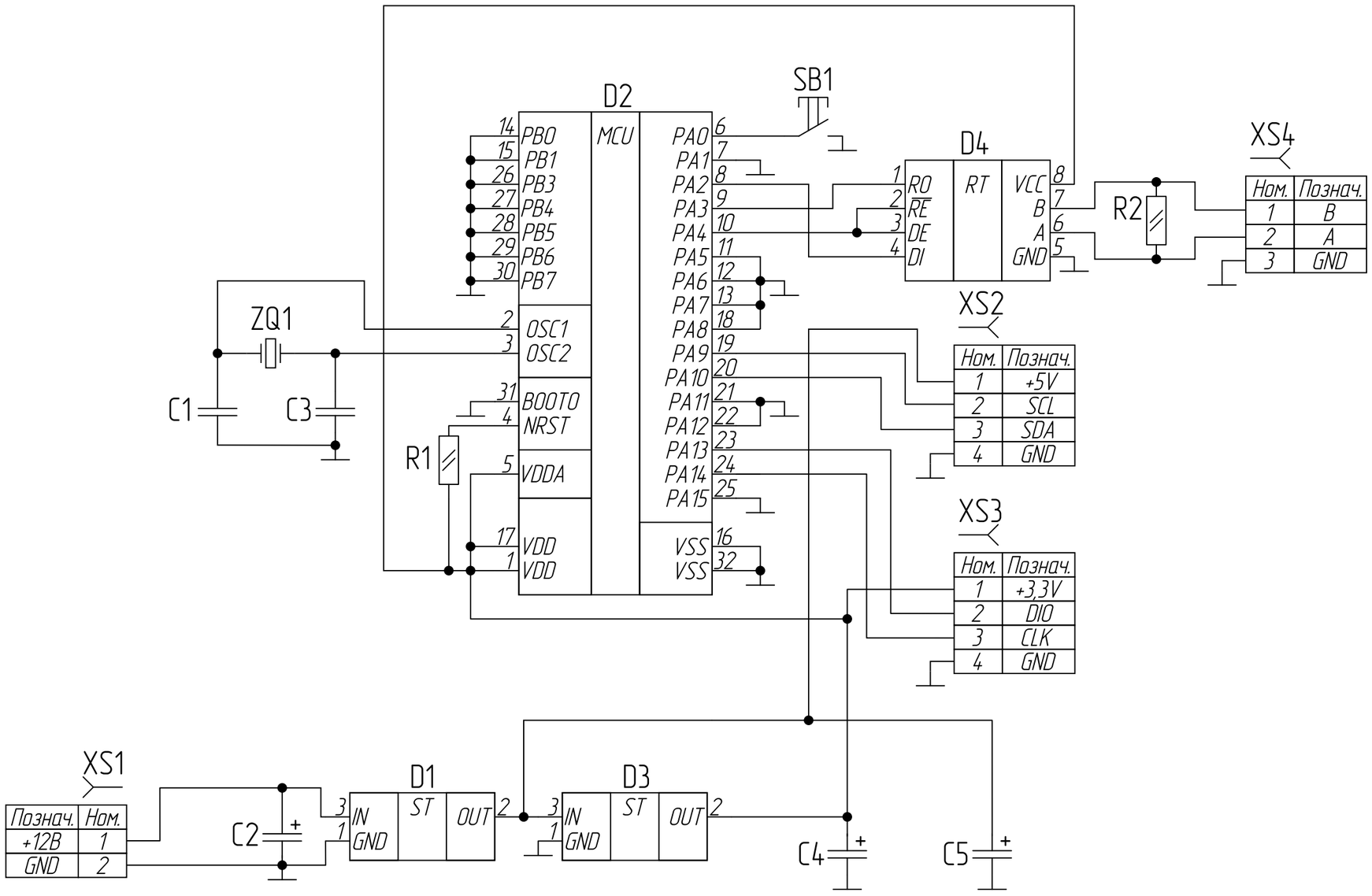
					КПТР.020051.01.03.Е1			
					Засіб автоматизованого захисту від несанкціонованого спостереження Схема електрична структурна	Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата		Н		
Розроб.		Разовий О. О.				Аркуш 1	Аркушів 1	
Перев.		Петрушак В. С.						
Т. контр.						ХНУ,ФІТ		
Н. контр.		Пивовар О. С.						
Затв.		Підченко С. К.						

1. Вивід 5 D1 підключити до кола V+
2. Вивід 2 D1 підключити до кола GND



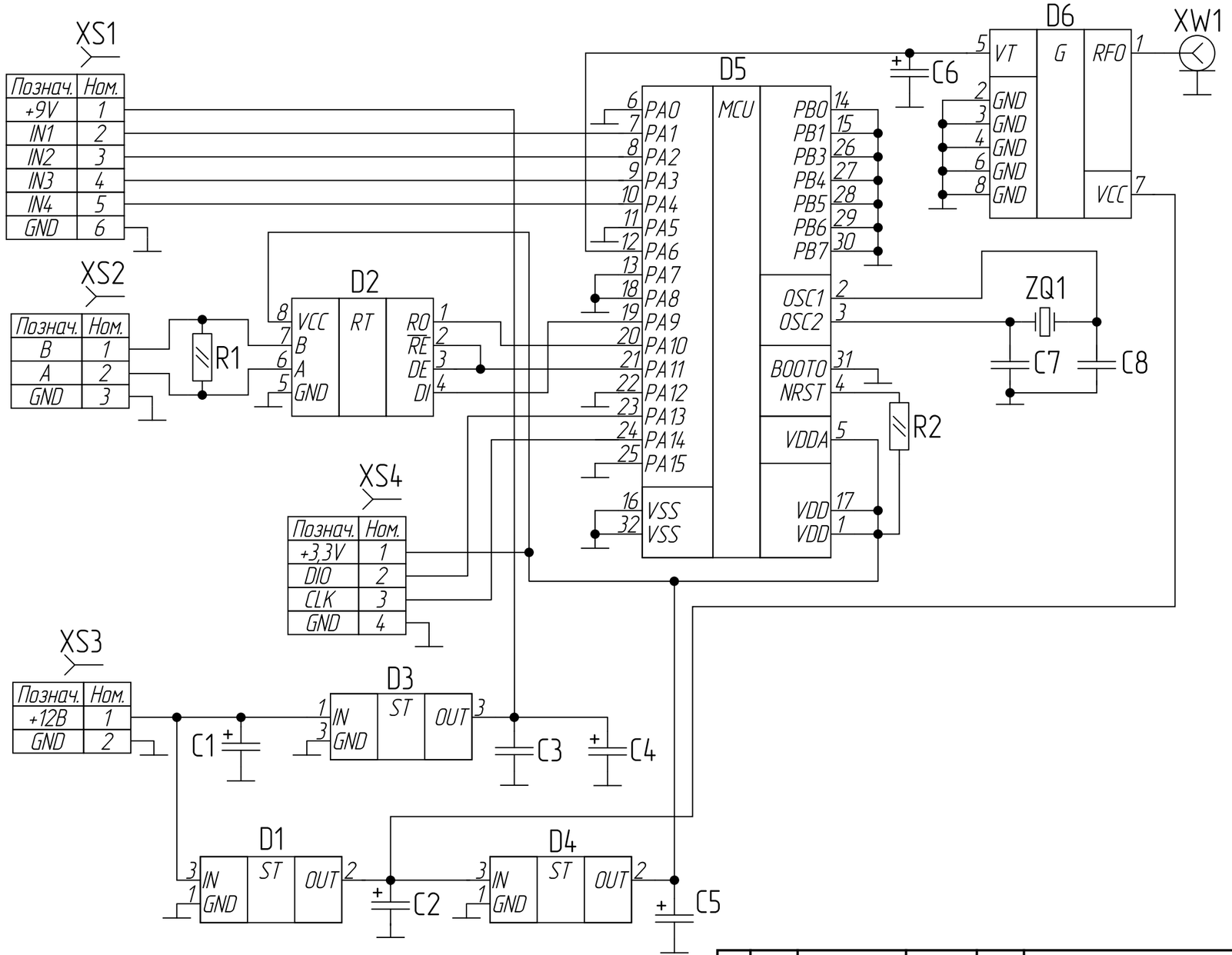
					КПТР.020051.01.03.Е3			
					Засіб автоматизованого захисту від несанкціонованого спостереження Схема електрична принципова	Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата		Н		
Розроб.		Разовий О. О.				Аркуш 1	Аркушів 3	
Перев.		Петрушак В. С.						
Т. контр.						ХНУ,ФІТ		
Н. контр.		Пивовар О. С.						
Затв.		Підченко С. К.						

КПТР.020051.01.03.Е3



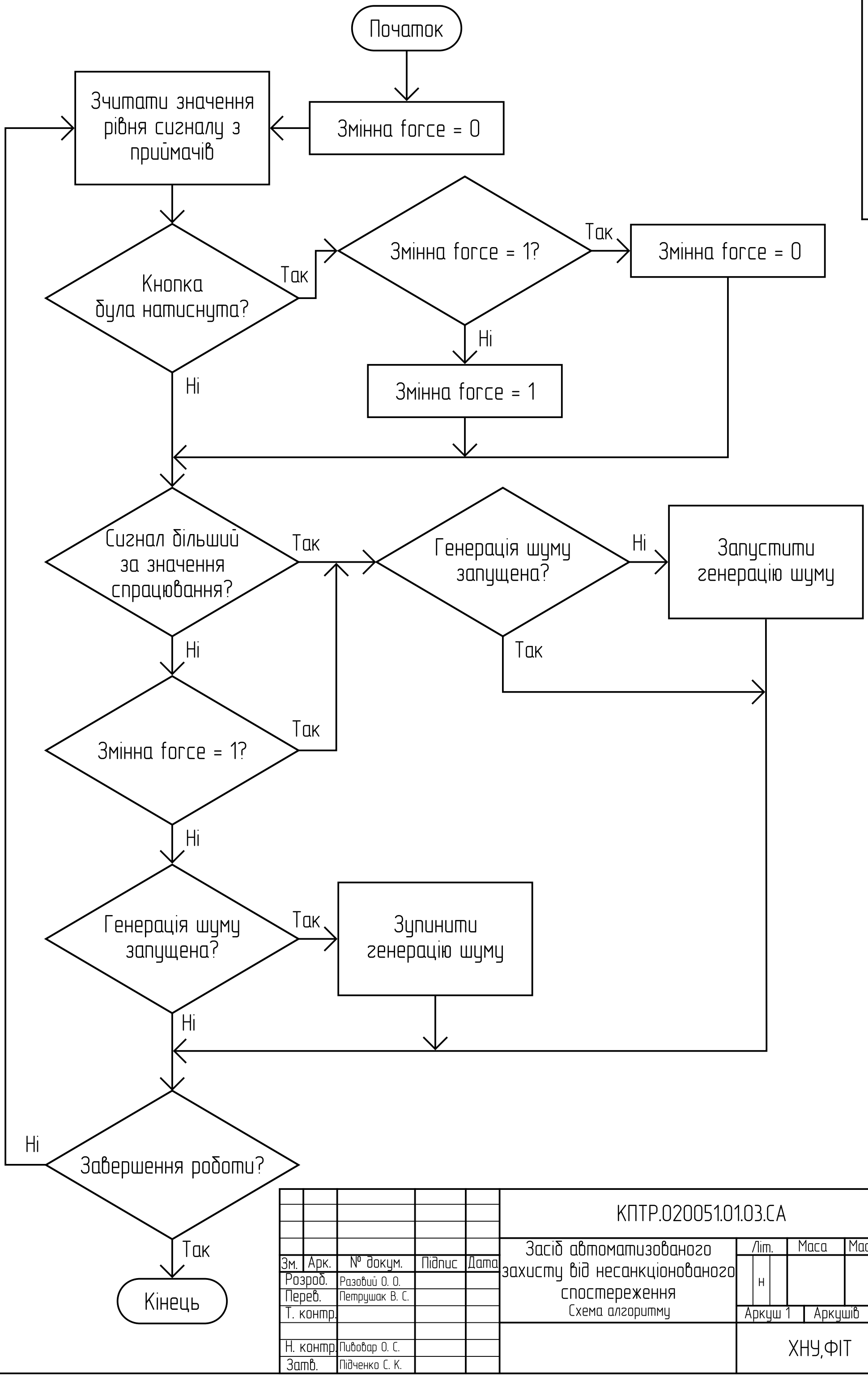
					КПТР.020051.01.03.Е3		
Зм.	Арк.	№ докум.	Підпис	Дата	Засіб автоматизованого захисту від несанкціонованого спостереження Схема електрична принципова	Літ.	Маса / Масштаб
Розроб.		Разовий О. О.				Н	
Перев.		Петрушак В. С.				Аркуш 2	Аркушів 3
Т. контр.							
Н. контр.		Пивовар О. С.				ХНУ,ФІТ	
Затв.		Підченко С. К.					

КПТР.020051.01.03.Е3



					КПТР.020051.01.03.Е3			
					Засіб автоматизованого захисту від несанкціонованого спостереження Схема електрична принципова	Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата		Н		
Розроб.		Разовий О. О.				Аркуш 3	Аркушів 3	
Перев.		Петрушак В. С.						
Т. контр.						ХНУ,ФІТ		
Н. контр.		Пивовар О. С.						
Затв.		Підченко С. К.						

КПТР.020051.01.03.СА



					КПТР.020051.01.03.СА	
Зм.	Арк.	№ докум.	Підпис	Дата	Засіб автоматизованого захисту від несанкціонованого спостереження Схема алгоритму	Літ. / Маса / Масштаб: Н
Розроб.		Разовий О. О.				Аркуш 1 / Аркушів 1
Перев.		Петрушак В. С.				
Т. контр.						ХНУ,ФІТ
Н. контр.		Пивовар О. С.				
Затв.		Підченко С. К.				

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 14%**

ID: 129194 Назва: Засіб автоматизованого захисту від несанкціонованого спостереження Додано в БД: 2024-06-09 Автора: Разовий Олександр Олегович Керівники: Петрушак Володимир Степанович Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	51046	754	1331 (3%)	24 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Kafedra TMIT KhNU

ID перевірки:
1016339776

Дата перевірки:
09.06.2024 23:07:05 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
09.06.2024 23:16:46 EEST

ID користувача:
100005657

Назва документа: Разовий_ТР2-20-1

Кількість сторінок: 74 Кількість слів: 11020 Кількість символів: 77267 Розмір файлу: 3.47 MB ID файлу: 1016140927

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.07%
Схожість

Найбільша схожість: 2.13% з Інтернет-джерелом (https://pastebin.com/cz7b2Bbu)

7.15% Джерела з Інтернету | 746 ..... Сторінка 76

3.86% Джерела з Бібліотеки | 178 ..... Сторінка 81

0.05% Цитат

Цитати | 1 ..... Сторінка 82

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи | 7

Підозріле форматування | 26 сторінок

РІШЕННЯ КАФЕДРИ

ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОГО ПРОЕКТУ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Засіб автоматизованого захисту від несанкціонованого спостереження

Автор: **Разовий Олександр Олегович**

Спеціальність: **172 Телекомунікації та радіотехніка**

Освітня програма: Телекомунікації та радіотехніка

Науковий керівник: **к.т.н., доц. Петрушак Володимир Степанович**

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом(далі-зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	**Відповідає**
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження: Запозичення у розмірі 9.07%, виявлені в роботі відповідають тексту стандартних бланків, решта запозичень є випадковими, тому ці запозичення не є плагіатом, бо вони не стосуються практичної значущості роботи.

9. 06 .2024р.

Науковий керівник
к.т.н.,доц. Петрушак В.С.

Відповідальний за перевірку на плагіат
к.т.н.,доц. Пивовар О.С.

Зав.каф. ТМІТ
д.т.н., доц. Підченко С.К.

Завідувачу кафедри телекомунікацій,
медійних та інтелектуальних технологій ХНУ
Підченку Сергію Костянтиновичу
здобувача вищої освіти
студента 4 курсу, гр. ТР2-20-1
Разового Олександра Олеговича

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу мого кваліфікаційного проєкту виконаного за темою «Засіб автоматизованого захисту від несанкціонованого спостереження» для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою

07.06.2024 _____ Олександр РАЗОВИЙ

дата підпис

В І Д Г У К

на кваліфікаційний проект студента групи ТР2-20-1

Разового Олександра Олеговича

«Засіб автоматизованого захисту від несанкціонованого спостереження»

Розроблений засіб автоматизованого захисту від несанкціонованого спостереження базується на використанні радіотехнічного виявлення відеосигналів FPV-дронів та сигналів керування ними. Основними особливостями пристрою є відокремлений пульт керування, що дозволяє слідкувати за рівнем сигналу та керувати пристроєм на відстані, а також напруга живлення 12 вольт, що робить можливою роботу пристрою від бортової мережі автомобіля чи від акумулятора.

Під час виконання кваліфікаційного проекту студент Разовий О. О., з належною наполегливістю віднісся до вирішення поставлених завдань, зарекомендував себе кваліфікованим спеціалістом в області телекомунікацій та радіотехніки з глибокими системними теоретичними знаннями та добрими практичними навичками.

В цілому кваліфікаційний проект Разового Олександра Олеговича "Засіб автоматизованого захисту від несанкціонованого спостереження" відповідає вимогам до кваліфікаційних проектів та заслуговує на оцінку "відмінно", а сам автор – на присвоєння кваліфікаційного рівня бакалавр зі спеціальності 172 – "Телекомунікації та радіотехніка".

Науковий керівник _____ / Пономаренко Д.С. /

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Разовий Олександр Олегович

Тема: Засіб автоматизованого захисту від несанкціонованого спостереження

Спеціальність: 172 Телекомунікації та радіотехніка.

Обсяг кваліфікаційного проекту

Кількість листів креслень 5 Кількість сторінок пояснювальної записки 60 .

1.Короткий зміст кваліфікаційного проекту та прийнятих рішень: Кваліфікаційний проект "Засіб автоматизованого захисту від несанкціонованого спостереження" присвячений актуальній у теперішній час проблемі несанкціонованого спостереження з використанням FPV-дронів і є корисним в практичному застосуванні .

2.Висновок про відповідність кваліфікаційного проекту завданню: Зміст кваліфікаційного проекту повністю відповідає завданню .

3.Характеристика виконання кожного розділу: У першому розділі представлено огляд принципів роботи аналогічних пристроїв, що існують на ринку та огляд сфер застосування. У другому розділі приводиться розробка структурної схеми, та вибір основних компонентів пристрою. У третьому розділі представлено розробку принципової схеми і розрахунок її елементів. У четвертому розділі наведено алгоритм роботи пристрою, описуються робота програмного коду та інструменти, що були використані при розробці програмного забезпечення.

4.Позитивні сторони кваліфікаційного проекту: 1. Проведено огляд принципів роботи аналогічних пристроїв для захисту та протидії несанкціонованому спостереженню. Виділено їх особливості та вказані переваги і недоліки. 2. Розроблено структурну схему пристрою, зроблено вибір компонентів. 3. Представлено розробку принципової схеми і розрахунок її елементів. 4. Розроблено алгоритм роботи пристрою та програмне забезпечення, робота якого детально описана. Також розглянуто інструменти, що були використані при розробці програмного забезпечення.

5.Негативні сторони кваліфікаційного проекту: Серед недоліків роботи можна відмітити недостатньо розгорнутий огляд принципів роботи аналогічних пристроїв. Крім того по

тексту пояснювальної записки наявні орфографічні помилки

.

6.Оцінка графічного оформлення та пояснювальної записки кваліфікаційного проекту: З точки зору оформлення кваліфікаційний проект представлений п'ятьма графічними кресленнями і пояснювальною запискою обсягом 60 аркушів, що складається з чотирьох розділів. Оформлення пояснювальної записки знаходиться на належному рівні, послідовність викладення матеріалу є логічною та зрозумілою. Графічне оформлення виконано відповідно до теми кваліфікаційного проекту. Пояснювальна записка оформлена згідно вимог чинних стандартів

.

7.Відгук про кваліфікаційний проект в цілому: Виконаний проект відповідає загальним вимогам, що пропонуються до кваліфікаційних проектів бакалавра

.

8.Інші зауваження: Немає .

9.Оцінка кваліфікаційної роботи: Розглянувши представлений проект, вважаю що робота заслуговує оцінки "відмінно", а Разовий Олександр Олегович – присвоєння кваліфікації бакалавра зі спеціальності 172 Телекомунікації та радіотехніка.

10. Рецензент (прізвище, ім'я, по-батькові, місце роботи) Федуна Микола Васильович, доц. каф. АКІТ

«07» 06 2024р. підпис