

Детектор Плагиата v. 2961 - Отчёт оригинальности: 14.06.2026 11:53:34

Проанализированный документ: Диплом_Бородавка.docx Лицензия: ВОЛОДИМИР МАТІЄВСЬКИЙ

🔍 Тип поиска: Поиск переписанного 🔍 Язык: Uk

🔍 Тип проверки: Интернет

ТЭЕ и кодировка: DocX n/a

Детальный анализ тела документа:

🔍 Диаграмма соотношения частей:

Плагат 0% Оригинал 95.12%
Кавычки 0.79% ИИ 4.09%



🔍 Граф распределения зон:



🔍 Источники плагиата: 1

 0% 1 1. <https://vodiy.ua/znaky/7/7.1.1/>

🔍 Детали обработанных ресурсов: 143 - ОК / 13 - Ошибок

🔍 Важные замечания:

Википедия:



[не обнаружено]

Google Книги:



[не обнаружено]

Сервисы платных работ:



[не обнаружено]

Античит:

**Обнаружено
сокрытие!**

🔍 Античит-отчет UACE:

- Статус: Анализатор **Включен** Нормализатор **Включен** сходство символов установлено на **100%**
- Обнаруженный процент загрязнения UniCode: **12,6%** с лимитом: 4%
- Процент нераспознанных символов после нормализации: **7,4%**
- Все подозрительные символы будут отмечены фиолетовым цветом: **Abcd...**
- Найдены невидимые символы: 0

Рекомендации по оценке:

Особое внимание следует уделить анализу этого отчета! Предполагается, что этот документ содержит значительное количество символов, чуждых языку документа. Это прямое указание на то, что автор документа использовал специальное программное обеспечение(онлайн-веб-сервис, чтобы эффективно скрыть текст в попытке избежать обнаружения потенциального плагиата. Настоятельно рекомендуется передать это дело на более высокий уровень! В случае сомнений обращайтесь в службу поддержки Детектора плагиата!

Алфавитная статистика и анализ символов:

🔍 Активные ссылки (URL-адреса, извлеченные из документа):

URL не найдены

🔍 Исключённые ресурсы:

URL не найдены

🔍 Включённые ресурсы:

URL не найдены

Детальний аналіз документа:

Міністерство освіти і науки України ДЗ

Цитування: 0,01%

id: 1

«луганський Національний університет імені тараса шевченка»

Факультет технологій та інформаційних систем (назва факультету, інституту)
Інформаційних технологій та систем (назва кафедри) Пояснювальна записка до
дипломного проєкту (роботи) БАКАЛАВРА (освітньо-кваліфікаційний рівень) на тему:
РОЗРОБКА МАКЕТУ АКУСТИЧНОЇ СИСТЕМИ ВИЯВЛЕННЯ ПЕРЕШКОД ЗА ДОПОМОГОЮ
УЛЬТРАЗВУКОВОГО ДАЛЕКОМІРА [HC-SR 04 ARDUINO](#) Виконав: студент 4 курсу, групи____
напряму підготовки (спеціальності) 123

Цитування: 0%

id: 2

«Комп'ютерна інженерія»

(шифр і назва напряму підготовки, спеціальності) Бородавка Р. В. (прізвище та ініціали)
Керівник Донченко В. Ю. (прізвище та ініціали) Рецензент ____ Козуб Ю. Г. (прізвище та
ініціали) Лубни – 2026 року ЗМІСТ Змн. Арк. № докум. Підпис Дата Арк. 2 ІТС.4КІ.0126.03-ПЗ
Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф.
Семенов М.А. ЗМІСТ Літ. Акрушів 2 ЛНУ Кафедра ІТС, Гр.4КІ . ВСТУП РОЗДІЛ 1. ТЕОРЕТИЧНІ
ОСНОВИ 6 1.1. Загальний принцип роботи ультразвукових датчиків 6 1.2. Мікроконтролери
[Arduino](#) 7 1.3. Принцип роботи світлодіодів 7 1.4. Принцип роботи бузера 8 РОЗДІЛ 2. ОГЛЯД
УЛЬТРАЗВУКОВИХ СЕНСОРІВ, ЇХ АНАЛОГІВ ТА НАЯВНИХ РІШЕНЬ 10 2.1. Застосування
ультразвукових сенсорів в системах автоматизації 10 2.2. Порівняння існуючих технологій
та їх недоліки 11 РОЗДІЛ 3. АПАРАТНА ЧАСТИНА ПРОЄКТУ 16 3.1. Вибір компонентів 16 3.2.
Створення схеми підключення 17 3.3. Обґрунтування інженерних рішень 18 РОЗДІЛ 4.
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ 19 4.1. Складова програмного забезпечення 19 4.2. [Module.h](#) 20
4.3. Функції у файлі [Module.c](#) 21 4.4. Функції у файлі [Interact.cpp](#) 23 4.5. Скетч для [Arduino](#) 27
РОЗДІЛ 5. ТЕСТУВАННЯ ТА УСУНЕННЯ ПОМИЛОК 30 5.1. Виявлення та виправлення
апаратних недоліків 30 5.2. Виявлення та виправлення недоліків у коді 30 5.3. Ідеї для
покращення після проведеного тестування 31 ВИСНОВКИ 32 СПИСОК ВИКОРИСТАНОЇ
ЛІТЕРАТУРИ 34 ДОДАТОК А 36 ДОДАТОК Б 43 Змн. Арк. № докум. Підпис Дата Арк. 3
ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ ВСТУП Змн. Арк. № докум. Підпис
Дата Арк. 4 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз.
Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. ВСТУП Літ. Акрушів 2 ЛНУ Кафедра ІТС, Гр.4КІ .
Якщо провести аналіз автомобілів популярних в нашій країні, то можна дійти до висновку,
що українці їздять на відносно старих авто. Згідно з даними Інституту досліджень
авторинку: середній вік легкової машини в державі становить 16,6 років. Найпопулярніші
моделі, що імпортуються як вживані ([Volkswagen Golf](#), [Skoda Octavia](#), [Renault Megane](#)), у
базових комплектаціях 2000–2010-х років не оснащувалися парктроніком [3]. Ця система
пропонувалася лише як опція у дорожчих версіях, які рідко потрапляють на вторинний
ринок. Відповідно, більшість автомобілів, що експлуатуються в Україні, не мають
заводських систем допомоги при паркуванні. Також, не слід забувати, що в наших реаліях,
особливо в менших містах, залишилися радянські, вітчизняні та російські автомобілі.
Виявити їх відсоткову частку не так легко, оскільки статистика ведеться за
перереєстрацією, отже вона навіть приблизно не показує справжню кількість таких машин
через те, що їх власники рідко змінюються та часто купують такий транспорт без
переоформлення. Очевидно, що в таких авто немає ніякої допомоги при паркуванні. Більше
того, велика кількість з них, спроектована в старі часи, коли капот часто обмежував
видимість водієві, що лише ускладнює процес паркування. Окрім легкових автомобілів, не
слід забувати про спеціальний транспорт. З ним проблема паркування стоїть ще гостріше,
яка викликана великими габаритами та низькою оглядовістю. Тут також, середній вік є
дуже поважним. Як свідчить той же Інститут досліджень авторинку, середній вік
вантажівок становить 18 років [4]. З огляду на середній вік найпопулярніших моделей
спецтранспорту (понад 15–20 років), можна стверджувати, що більшість із них не
обладнані парктроніками [4]. Ця технологія поширилася лише після 2010 року, переважно
на легкових автомобілях та нових поколіннях комерційних фургонів. Старі вантажні моделі
(КамАЗ, [L1](#), [T1/T2](#)) практично гарантовано не мають таких систем. Як можна переконалися
самостійно, відвідавши автопарк комунального підприємства малого міста, що присутня
велика кількість радянських вантажівок (ЗІЛ, ГАЗ, КамАЗ, МАЗ). Змн. Арк. № докум. Підпис
Дата Арк. 5 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Якщо проаналізувати
все сказане вище, то легко побачити проблему. Більшість авто в Україні не мають системи
допомоги з паркуванням. Ціллю цієї дипломної роботи є створення доступної, відкритої,
модульної системи допомоги в паркуванні у гаражі, що повинна полегшити цю задачу для
різних категорій водіїв. Завданнями для створення такої системи є: Розробка схеми
прототипу системи. Реалізація алгоритмів обробки даних від ультразвукових сенсорів.
Реалізація подачі інформації від сенсорів водію. Тестування функціональності прототипу.
Результатом виконання цих завдань має бути стаціонарна модульна, відкрита та доступна
система допомоги в паркуванні. РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ 1.1. Загальний принцип
роботи ультразвукових датчиків Ультразвукові датчики – це вимірювальні пристрої, які
знаходять широке застосування в різних галузях, від робототехніки та автоматизації до
медицини та автомобільної промисловості. Вони дозволяють вимірювати відстань до

об'єктів безконтактним способом, використовуючи принцип ехолокації [1]. Саме в цій роботі, буде їх використано для створення системи допомоги в парковці автомобілів. Ультразвуковий датчик випромінює короткі імпульси ультразвукових хвиль, які нечутні для людського вуха. Коли ці хвилі зустрічають перешкоду, вони відбиваються назад до датчика. Датчик вимірює час, який пройшов між випромінюванням імпульсу та прийомом відбитого сигналу (ехо) [1]. Детальніше спосіб обчислення відстані буде розібрано в розділі 3. Такі розрахунки як: вимірювання відстані, обчислення точності датчики, кореляція вологи повітря на швидкість звуку, розрахунок затримки датчиків — будуть наведені у Додатках. Більшість ультразвукових датчиків мають чотири контакти: [VCC](#), [GND](#), [Trigger](#) та [Echo](#). [VCC](#) підключається до джерела живлення (зазвичай 5В), [GND](#) — до землі. [Trigger](#) використовується для запуску випромінювання ультразвукового імпульсу, а [Echo](#) — для отримання відбитого сигналу [1]. Для роботи з ультразвуковим датчиком необхідно написати програму, яка буде керувати [Trigger](#) та [Echo](#) пінами [1]. Зазвичай, спочатку на [Trigger](#) пін подається короткий імпульс (близько 10 мікросекунд), потім програма переходить в режим очікування відбитого сигналу на [Echo](#) пін. Тривалість цього сигналу пропорційна відстані до об'єкта [1]. Проте в нашому випадку, для збільшення точності вимірюванням, запобігаючи наводкам від інших датчиків, програмно буде реалізовано почергове опитування ультразвукових сенсорів, за Змн. Арк. № докум. Підпис Дата Арк. 6 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ Літ. Акрушів 4 ЛНУ Кафедра ІТС, Гр.4КІ . допомогою бібліотеки [NewPing](#). Також буде впроваджено механізми фільтрації значень, засобами цієї бібліотеки. Про це детальніше буде описано в розділі 3. 1.2. Мікроконтролери [Arduino Arduino](#) — це відкрита апаратно-програмна платформа, створена для швидкого прототипування електронних систем. Вона виникла у 2005 році в Італії як інструмент для студентів, які не мали глибоких знань у програмуванні чи електроніці, але хотіли створювати інтерактивні проекти. Основна ідея полягає в тому, що користувач отримує просту у використанні мікроконтролерну плату та середовище програмування, яке дозволяє писати невеликі програми — так звані

Цитування: 0%

id: 3

«скетчі»

— і завантажувати їх на плату через [USB](#). [Arduino](#) може зчитувати сигнали від різних сенсорів, наприклад, ультразвукових, температурних чи світлових, і відповідно керувати виконавчими пристроями — світлодіодами, моторами, дисплеями або звуковими індикаторами. Завдяки відкритій архітектурі та великій спільноті користувачів, платформа підтримує безліч додаткових модулів, таких як [Wi-Fi](#), [Bluetooth](#) чи [GSM](#), що робить її універсальною для проєктів у сфері робототехніки, автоматизації, інтернету речей та навіть мистецьких інсталяцій. 1.3. Принцип роботи світлодіодів Світлодіод працює на основі явища електролюмінесценції, коли електрони та дірки в напівпровідниковому матеріалі рекомбінують у зоні [p-n](#) переходу. У момент прикладання прямої напруги електрони переходять із зони провідності в зону валентності, і ця енергетична різниця вивільняється у вигляді фотона — тобто світла. Колір випромінювання визначається шириною забороненої зони матеріалу: чим вона більша, тим коротша довжина хвилі і тим ближче світло до синього спектра; при меншій ширині забороненої зони випромінювання зміщується до червоного. Для отримання білого світла використовують синій світлодіод у поєднанні з люмінофором, який перетворює частину випромінювання в жовтий спектр, і разом вони створюють відчуття білого світла. Заборонена зона — це енергетичний проміжок між валентною зоною та зоною провідності в напівпровіднику. У валентній зоні перебувають електрони, Змн. Арк. № докум. Підпис Дата Арк. 7 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ які беруть участь у хімічних зв'язках, а зона провідності — це рівні енергії, де електрони можуть вільно рухатися й переносити струм.

AI generated text detected: ChatGPT 5.3, : 54,75%

id: 4

Якщо електрон отримує достатньо енергії, він переходить із валентної зони в зону провідності, залишаючи після себе «дірку». У світлодіоді саме величина цієї забороненої зони визначає колір світла: коли електрон повертається назад і рекомбінує з діркою, він вивільняє енергію у вигляді фотона. Чим більша ширина забороненої зони, тим вища енергія фотона і коротша довжина хвилі, тобто світло зміщується до синього спектра. При меншій ширині забороненої зони випромінювання має меншу енергію й відповідає червоному або жовтому світлу. Таким чином, заборонена зона є ключовим параметром, що визначає спектральні властивості світлодіода. Отже, принцип роботи світлодіода полягає у прямому перетворенні електричної енергії на світлову без проміжних процесів нагрівання чи газового розряду, що робить його набагато ефективнішим і довговічнішим за традиційні джерела світла. 1.4. Принцип роботи бузера Бузер, або електричний дзижчок, є різновидом звукового сигналізатора, який використовується для створення характерного різкого звуку. Він складається з корпусу з мембраною чи котушкою, що починає коливатися під дією електричного струму, перетворюючи електричну енергію на акустичні хвилі.

Такі пристрої застосовуються у побутовій техніці, транспорті, системах сигналізації та різних електронних схемах, де звук слугує способом комунікації з користувачем — наприклад, повідомляє про завершення процесу, несправність чи низький заряд батареї [2]. Існує кілька основних типів бузерів. Електромеханічний за принципом дії нагадує

електричний дзвінок, але має спрощену конструкцію без молоточка. Електромагнітний працює подібно до маленького динаміка: котушка створює змінне магнітне поле, яке змушує рухатися магніт і прикріплену до нього металеву мембрану, що генерує звук. П'єзоелектричний бузер базується на п'єзоелектричному ефекті — пластина з п'єзоелектричного матеріалу деформується під дією струму, викликаючи коливання мембрани та утворення Змн. Арк. № докум. Підпис Дата Арк. 8 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ звукових хвиль [2]. За способом керування бузери поділяють на активні та пасивні. Активні мають вбудований генератор сигналу, тому для роботи достатньо подати напругу, і вони самостійно створюють звук певної частоти. Пасивні ж не мають генератора і потребують зовнішнього сигналу, який вони лише перетворюють у звук. Активні моделі більш універсальні, дозволяють регулювати параметри звучання, тоді як пасивні простіші й застосовуються там, де потрібне лише відтворення зовнішнього сигналу [2]. Вибір конкретного бузера залежить від його параметрів — частоти, гучності, напруги живлення, імпедансу, типу звуку та умов експлуатації. Саме ці характеристики визначають, наскільки пристрій буде ефективним у конкретному застосуванні, чи то побутова техніка, автомобільна електроніка, чи системи безпеки [2]. Змн. Арк. № докум. Підпис Дата Арк. 9 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ РОЗДІЛ 2. ОГЛЯД УЛЬТРАЗВУКОВИХ СЕНСОРІВ, ЇХ АНАЛОГІВ ТА НАЯВНИХ РІШЕНЬ 2.1. Застосування ультразвукових сенсорів в системах автоматизації Попит на системи автоматизації паркування в замкнених просторах, таких як приватні гаражі, криті паркінги чи автомобільні бокси, стабільно зростає разом із загальним розвитком, про що свідчать відгуки та їх кількість на помічників у паркуванні [11]. Ультразвукова локація є найбільш виправданим вибором для безконтактного вимірювання відстані. На відміну від оптичних чи інфрачервоних сенсорів, ультразвукові прилади не залежать від рівня освітленості приміщення, кольору кузова автомобіля чи наявності пилу й туману в повітрі, забезпечуючи стабільну точність вимірювань у діапазоні від кількох сантиметрів до кількох метрів. [10] Сучасний ринок пропонує ряд готових комерційних рішень для стаціонарного настінного монтажу, проте детальний аналіз їхньої архітектури показує суттєвий розрив між можливостями та вартістю. Найбільш поширеними є моноблочні світлофорні індикатори, яскравим прикладом яких є пристрій [STKR Concepts Adjustable Garage Parking Sensor](#) (рис. 2.1) [5] та його масові аналоги типу [Zone Tech Car Parking Sensor](#) (рис 2.2) [6]. Ці системи монтуються на фронтальну стіну гаража і використовують триколірну світлодіодну матрицю, де зелений колір означає початок руху в зоні видимості, жовтий — необхідність зниження швидкості, а червоний — зупинку у визначеній користувачем точці. Змн. Арк. № докум. Підпис Дата Арк. 10 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. РОЗДІЛ 2. ОГЛЯД УЛЬТРАЗВУКОВИХ СЕНСОРІВ, ЇХ АНАЛОГІВ ТА НАЯВНИХ РІШЕНЬ Літ. Акрушів 6 ЛНУ Кафедра ІТС, Гр.4КІ . Рис. 2.1. [STKR Concepts Adjustable Garage Parking Sensor](#). Рис 2.2. [Zone Tech Car Parking Sensor](#). 2.2. Порівняння існуючих технологій та їх недоліки Порівняння існуючих технологічних підходів дозволяє чітко виявити проблеми відомих рішень, такі як низька точність за наявності зовнішніх перешкод, висока вартість компонентної бази, складність монтажу в умовах реального гаражного боксу, а також повна відсутність можливостей для масштабування та програмної модифікації. Прикладом практичного застосування інфрачервоної технології в системах контролю простору є оптичні датчики бар'єрного типу, такі як [Chamberlain LiftMaster CPS-U](#) (рис. 2.3) [7], що широко використовуються в автоматизації гаражних воріт. Проте головним обмеженням таких систем є їхня бінарна логіка роботи: вони здатні зафіксувати лише сам факт перетину світлового променя, але не дозволяють відстежувати динаміку наближення автомобіля до перешкоди. Спроби інтегрувати аналогічні інфрачервоні далекомірні модулі, наприклад [Sharp GP2Y0A21YK0F](#) (рис. 2.4) [8], у паркувальні асистенти виявили Змн. Арк. № докум. Підпис Дата Арк. 11 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ їхню високу вразливість до зовнішніх чинників. Зокрема, пряме сонячне світло з боку відкритих воріт може просто

Цитування: 0%

id: 5

«засліпити»

фотоприймач, викликаючи збої. Крім того, точність вимірювання фінальної координати тут суттєво залежить від кольору та чистоти кузова автомобіля. Темне або забруднене лакофарбове покриття активно поглинає інфрачервоні хвилі, через що датчик спрацьовує із запізненням, підвищуючи ризик зіткнення. Додатковим деструктивним фактором є пил, природний для будь-якого гаража: він швидко вкриває оптичні лінзи нальотом, що змушує користувача постійно протирати й обслуговувати прилад. Рис 2.3. Інфрачервоний сенсор бар'єрного типу [Chamberlain LiftMaster CPSU garage safety sensor](#) Рис 2.4. Інфрачервоний далекомірний модуль [Sharp GP2Y0A21YK0F](#) Змн. Арк. № докум. Підпис Дата Арк. 12 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Рис 2.5. Лазерний покажчик позиціонування [Maxsa Innovations Park Right Laser Parking Guide](#) Альтернативний технологічний підхід реалізовано в лазерних покажчиках позиціонування, яскравим представником яких є пристрій [Maxsa Innovations Park Right Laser Parking Guide](#) (рис 2.5) [9]. Цей прилад проектує тонкий лазерний промінь на капот або панель приладів автомобіля, вказуючи водієві на досягнення потрібної точки зупинки. Незважаючи на високу точність самого світлового променя, експлуатація такої системи супроводжується значними складнощами під час інсталяції. Прилад вимагає дуже точного механічного калібрування

кута нахилу випромінювача під конкретну модель автомобіля та ріст водія. Найменша зміна траєкторії в'їзду або заміна транспортного засобу на модель з іншими габаритами вимагає повного ручного переналаштування системи. Окрім цього, вузькоспрямований лазерний промінь контролює лише одну геометричну точку, повністю ігноруючи бічні зони. Якщо водій змістить автомобіль праворуч або ліворуч, система не зможе попередити його про небезпечне наближення до стін, стелажів чи навісних полиць. Якщо заглянути всередину комерційних пристроїв на кшталт [STKR Concepts Adjustable Garage Parking Sensor](#), ми побачимо там найпростішу електронну плату (мікроконтролер) та один ультразвуковий датчик. Купуючи такий прилад за 30–65 доларів США, користувач значною мірою переплачує за маркетинг і логістику виробника [5]. На противагу цьому, наш асистент паркування побудований на доступних і масових платформах [Arduino](#) або [ESP32](#), що дозволяє знизити базову вартість деталей у кілька разів та розширює можливості. Змн. Арк. № докум. Підпис Дата Арк. 13 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Крім фінансового боку, існують і суттєві конструктивні обмеження. Пристрої типу [STKR](#) зроблені як моноблок — тобто датчик і світлове табло

Цитування: 0%

id: 6

«світлофора»

жорстко об'єднані в одному корпусі. Через це водій може контролювати відстань лише в одній точці (наприклад, перед бампером). Якщо ж гараж має нестандартну форму, або всередині є виступаючі полиці й бічні опори, доводиться купувати кілька окремих комплектів [STKR](#). Оскільки вони працюють самі по собі й не пов'язані один з одним, стіни гаража швидко захаращуються купою дротів та блоків живлення. Наша розробка вирішує цю проблему завдяки модульній структурі: до однієї центральної плати можна легко підключити кілька датчиків для контролю машини з різних боків. Ще одна суттєва різниця полягає в тому, як саме пристрій спілкується з водієм. Моноблок [STKR](#) дає лише візуальні підказки через вбудовані світлодіоди. Проте на практиці, якщо в гараж падає яскраве сонячне світло або на пластиковий корпус осідає пил, розгледіти ці кольори стає важко. У нашому пристрої цей недолік повністю усунуто. Світловий сигнал завжди дублюється звуком. Це дозволяє водієві орієнтуватися на слух, дивитися виключно в дзеркала заднього виду і не відволікатися на настінні екрани, що робить процес паркування набагато безпечнішим. Змн. Арк. № докум. Підпис Дата Арк. 14 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Головним мінусом фабричних систем є їхня

Цитування: 0%

id: 7

«закритість».

Купуючи пристрій [STKR Concepts](#), ви отримуєте жорстко зашиту програму, яку неможливо змінити. Наша ж система має повністю відкритий вихідний код. Це дає повну свободу дій: користувач може самостійно змінити безпечну відстань під габарити саме свого автомобіля, налаштувати частоту звуку чи миготіння світлодіодів, запрограмувати захист від помилкових спрацювань (наприклад, якщо повз датчик пробіжить домашня тварина) або навіть інтегрувати систему в загальну мережу розумного будинку. Щоб реалізувати всі ці переваги без зайвих витрат, ми розділили систему на один головний блок (

Цитування: 0%

id: 8

«Master»

) та кілька простих периферійних вузлів (

Цитування: 0%

id: 9

«Slave»

). Замість того, щоб ставити дорогий електронний

Цитування: 0%

id: 10

«МОЗОК»

у кожен точку контролю (як це довелося б робити у випадку з купівлею кількох пристроїв [STKR](#)), у нашому рішенні за всю логіку, обробку сигналів та фільтрацію цифрового шуму відповідає лише один центральний мікроконтролер. Самі ж допоміжні модулі — це максимально прості й пластикові корпуса, на яких встановлено лише ультразвуковий датчик, світлодіод та бузер. Наша система виходить значно дешевшою за фабричні аналоги типу [STKR Concepts](#), але водночас пропонує набагато більшу надійність, гнучкість та можливість для підлаштування під потреби конкретного водія. Допоміжні модулі є максимально спрощеними та дешевими платами, які містять лише ультразвуковий датчик відстані та елементи локального сповіщення у вигляді світлодіода і бузер. Керування живленням цих індикаторів на периферійних платах реалізовано за допомогою пінів Ардуіно, що дозволяє програмно контролювати їх. Для кожного модуля виділено своє джерело живлення. Подібне апаратне рішення суттєво знижує теплове та електричне навантаження на стабілізатор напруги центральної плати, запобігаючи перегріву системи під час тривалої роботи індикаторів. Змн. Арк. № докум. Підпис Дата Арк. 15 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Кількість додаткових точок контролю обмежується лише кількістю вільних пінів введення-виведення на центральній платі та загальною потужністю джерела живлення. На початковому етапі розробки допоміжні модулі реалізуються у вигляді компактних макетних плат, які легко фіксуються

на стінах гаража. Такий підхід дозволяє створити надійну багатоточкову систему контролю геометрії паркування за мінімальної собівартості компонентів, забезпечуючи користувачеві можливість повної програмної адаптації пристрою під конкретні умови експлуатації.

РОЗДІЛ 3. АПАРАТНА ЧАСТИНА ПРОЄКТУ 3.1. Вибір компонентів Для реалізації такого проєкту, необхідно обрати правильні комплектуючі. Перш за все, потрібен мікроконтролер. Оскільки, мережевих операцій не передбачається, то [ESP12](#) чи [ESP32](#) є сумнівним вибором, оскільки вони вартують більше і не потрібні в данному проєкті. У них є безсумнівна перевага: апаратна багатозадачність, проте для реалізації системи допомоги в паркуванні — це не є обов'язковим. Буде використана псевдо багатозадачність, детальніше про це написано в четвертому розділі. Ідеальним вибором для системи допомоги в паркуванні є мікроконтролер [Arduino nano](#). Його ключовою перевагою є мініатюрний розмір, живлення від 5В, та низька вартість [12]. Важливо уточнити, що [Arduino](#) мають відкриту архітектуру, що дозволяє виробляти суттєво дешевші за оригінал репліки, які повністю відповідають вимогам цієї системи. Головним недоліком цього мікроконтролера — є недостатня кількість портів. Ця проблема може бути вирішена за допомогою [I2C](#) модуля розширення портів на чипі [MCP23017](#), або будь яким іншим модулем розширення портів. Для простоти монтажу та економії місця, були використані [rgb](#) світлодіоди із спільним анодом. Таким чином, вдалося замінити три світлодіода одним. Роз'єми для підключення модулів представляють з себе комп'ютерні [Molex Micro Fit](#). Кабелі обстискаються простим [Dupont](#) роз'ємом та вставляються в конектор. Такий вибір забезпечує простоту розробки, з недоліків можна визначити незручність в ремонті такого рішення, оскільки дістати провід з такого роз'єму дуже складно. Для передачі інформації використовується 8-pin [Molex Micro Fit](#), а для живлення - 4-pin. Для передачі звукового сигналу було обрано активний бузер. Такий вибір був спричинений тим, що пасивний бузер вимагав користування ним через функцію [tone\(\)](#). Це стало неможливим, оскільки бібліотека [NewPing.h](#) вже Змн. Арк. № докум. Підпис Дата Арк. 16 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. РОЗДІЛ 3. АПАРАТНА ЧАСТИНА ПРОЄКТУ Літ. Акрушів З. ЛНУ Кафедра ІТС, Гр.4КІ . використовує її, що призводить до помилки ще на моменті компілювання вихідного коду. Для обміну інформацією з датчиками, взаємодії зі світлодіодами та визначенню: чи підключено модуль, було обрано восьмижильний екранований кабель для сигналізації. Для забезпечення стабільності роботи системи було прийнято рішення використати додаткові жили маси в кабелі. Такий підхід має кілька технічних переваг. По-перше, зменшується сумарний опір провідника завдяки паралельному з'єднанню кількох жил. Це призводить до зниження падіння напруги та зменшення теплових втрат при передачі струму. По-друге, кожна сигнальна пара отримує власний шлях повернення струму, що зменшує електромагнітні наведення та покращує якість сигналу. Це особливо важливо для високочастотних систем, особливо при великій довжині проводів, де наведення можуть суттєво впливати на точність передачі даних.

AI generated text detected: ChatGPT 5.3, : 51,16%

id: 11

Таким чином, використання додаткових жил маси не є надлишковим, а навпаки — забезпечує підвищену надійність, зниження рівня шумів та стабільність роботи всієї системи. 3.2. Створення схеми підключення Перш за все, необхідно створити надійну схему підключення всіх приладів та роз'ємів.

Слід врахувати, що на реальному проєктові, в зв'язку із обмеженням інструментарієм, проводи можуть мати однаковий колір, або відрізнятися від показаних на схемі. Основною задачею схеми є показати загальний принцип роботи всього приладу. Змн. Арк. № докум. Підпис Дата Арк. 17 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Рис. 3.1. Схема підключення. 3.3. Обґрунтування інженерних рішень В двох майстер-модулі було прийнято рішення повністю відмовитися від пайки для простоти ремонту та зміни конфігурації. Це було досягнути шляхом використання [Wago](#) клемників для створення двох шин, однієї позитивної, іншої негативної. Таке вирішення мало один недолік: для такого підходу потрібно забезпечити більше місця в корпусі, що було виконано. В модулях без [Arduino](#), використовувалася пайка. Для тих же цілей, що і [Wago](#) конектори в майстер-модулі. Корпуси було обрано універсальні, пластикові. В них було створено отвори для датчиків та роз'ємів. Через те, що майстер-модуль займає більше місця, було неможливо його помістити в один корпус, тому було з'єднано два пластикових каркаса та створено отвір між ними для проводки. Для з'єднання між модулями, було обрано роз'єми [molex](#). В результаті, через ці роз'єми, було допущено помилку, що примусило провести заміну на інші. Змн. Арк. № докум. Підпис Дата Арк. 18 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ 4.1. Складова програмного забезпечення Програмну частину можна розділити на два компоненти: бібліотека та скетч для Ардуіно. Бібліотеку створено в

Цитування: 0%

id: 12

«чистому»,

функціональному стилю С.

AI generated text detected: ChatGPT 5.3, : 57,74%

id: 13

Оскільки, проєкт не великий, то немає потреби створювати бібліотеку, яка б використовувала усі можливості мови C++, а саме повноцінний ООП та додаткові можливості для абстракції. Вибір функціонального стилю чистого С дозволить зекономити

пам'ять, покращити синтаксис та спростити архітектуру бібліотеки. Завдяки такому підходу, бібліотеку легко розширювати, вносити зміни та використовувати. Інкапсуляцію було досягнуто простим внесенням методів у `.h` файл. Таким чином, ми створили примітивний інтерфейс, який дає доступ лише до деяких функцій. Нижче, в цьому розділі, будуть описані всі функції та сама модель. Бібліотека створена з використанням як блокуючого, так і неблокуючого коду. Для того, щоб реалізувати роботу всіх світлодіодів та бузерів незалежно, використовується неблокуючий код. На [Arduino](#), через відсутність як апаратної так і програмної багатопоточності, інженер вимушений застосовувати псевдоасинхронний підхід за допомогою функції `millis()`.

Реалізація полягає у застосуванні паттерну неблокуючого таймінгу. Якщо пояснити детальніше, то в [Arduino](#) є апаратний таймер, за допомогою якого, ми можемо дізнатися скільки мілісекунд пройшло від початку роботи мікроконтролера, за допомогою `millis()`, таймер скидається кожні 49,7–50 днів, в даному проєкті цим можна знехтувати. За допомогою цього, ми можемо зафіксувати певне значення часу, після цього, під час кожної ітерації `loop()`, віднімати його від поточного, і якщо різниця задовольнить вимоги до інтервалу, то виконувати певну дію, для прикладу: ввімкнути світлодіод, та потім таким же чином його вимкнути. Для збереження станів світлодіодів, бузерів та їх останнього застосування, у конструкті `Module` Змн. Арк. № докум. Підпис Дата Арк. 19 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ Літ. Акрушів 11 ЛНУ Кафедра ІТС, Гр.4КІ . передбачені спеціальні поля типу `UnsignedLong`. Скетч, як і більшість скетчів для [Arduino](#), складається з методів `setup()` та `loop()`. В методі `setup()` ми задаємо вхідні значення, як от номери пінів, а в `loop()` проводиться взаємодія з бібліотекою [NewPing.h](#), опитування датчиків та передача їх показників у функцію, що відповідає за взаємодію з світловою та звуковою індикаціями. Нижче, в цьому розділі, його буде описано детальніше.

4.2. `Module.h` Створена бібліотек реалізує модель `Module`. Вона забезпечує інкапсуляцію параметрів та функцій, необхідних для взаємодії з сенсорами та виконавчими пристроями. Структура `Module` містить набір полів, що відповідають за конфігурацію пінів та станів системи, тоді як допоміжна структура `ModulePins` створена для гнучкості коду та зручнішого менеджменту пінів. Вона не є обов'язковою, проте дозволяє зробити код чистішим, уніфікованим та більш адаптивним до змін. Таблиця 4.1 Змн. Арк. № докум. Підпис Дата Арк. 20 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Типи полів та їх призначення в структурі `Module` Поле Тип Призначення `triggerPin int` Пін для сигналу `Trig` ультразвукового сенсора `HC SR04` `echoPin int` Пін для сигналу `Echo` ультразвукового сенсора `HC SR04` `safePin int` Пін для підключення світлодіода, що вмикається коли об'єкт далеко `warningPin int` Пін для підключення світлодіода, що вмикається коли об'єкт близько `dangerPin int` Пін для підключення світлодіода, що вмикається коли об'єкт небезпечно близько `buzzerPin int` Пін для підключення бузера (звукової сигналізації) `signalPin int` Пін, що замикається на землю, коли підключено модуль. Використовується для динамічного визначення модулів Поле Тип Призначення `lightState int` Поточний стан світлової індикації `noiseState int` Поточний стан звукової індикації `lastBlink unsigned long` Час останнього миготіння світлодіода `lastNoise unsigned long` Час останнього звукового сигналу Таблиця 4.2 Типи полів та їх призначення в структурі `ModulePins` Поле Тип Призначення `triggerPin int` Пін для сигналу `Trig` ультразвукового сенсора `HC SR04` `echoPin int` Пін для сигналу `Echo` ультразвукового сенсора `HC SR04` `safePin int` Пін для підключення світлодіода, що вмикається коли об'єкт далеко `warningPin int` Пін для підключення світлодіода, що вмикається коли об'єкт близько `dangerPin int` Пін для підключення світлодіода, що вмикається коли об'єкт небезпечно близько `buzzerPin int` Пін для підключення бузера (звукової сигналізації) `signalPin int` Пін, що замикається на землю, коли підключено модуль. Використовується для динамічного визначення модулів В цьому ж файлі оголошені методи, до яких може мати доступ скетч. Таким чином реалізовано інтерфейс в функціональному стилі C, без використання ООП C++, яке не є тут необов'язковим. Функції, що наявні в `Module`, будуть описані в наступному підрозділі, що відповідає за `module.c` та `interact.cpp`.

4.3. Функції у файлі `Module.c` Першою функцією служить `findModulesCount`, яка повертає кількість підключених модулів у вигляді значення типу `int`. Змн. Арк. № докум. Підпис Дата Арк. 21 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Аргументи, що приймає функція: `ModulePins pins[]` – Служить посиланням на масив з пінами, по якому проходиться цикл, обирає звідти сигнальні піни та перевіряє їх стан за допомогою функції `myDigitalWrite()`. Якщо функція повертає значення `Low`, що означає замикаання піну, то виходить, що модуль підключено, отже додаємо його в загальну кількість модулів. Ця функція запускається одною із перших і є фундаментальною. За допомогою неї, скетч дізнається загальну кількість модулів та присвоює їм раніше визначені піни. Змн. Арк. № докум. Підпис Дата Арк. 22 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ `int size` – передаємо розмір масиву, що створюємо для збереження модулів. Визначає його скетч [Arduino](#), за допомогою `sizeof`. Ми повинні саме передати його в функцію, замість того щоб там визначити, оскільки інакше, в умові циклу не можемо вказати його розмір, тобто кінцеву точку для ітерацій. Визначити масив у функції неможливо, оскільки під час передачі масива у функцію, він перетворюється на вказівник на той масив, до якого ми не можемо застосувати `sizeof`. Функція `fillModules` виконує ініціалізацію масиву структур типу `Module` на основі даних, що передаються у вигляді масиву структур `ModulePins`. Її основне

призначення полягає у тому, щоб кожному модулю призначити відповідні апаратні ресурси (пін-коди мікроконтролера), а також встановити початкові значення внутрішніх змінних стану. Функція приймає три аргументи: масив `modules[]`, який містить об'єкти типу `Module`; параметр `count`, що визначає кількість модулів для обробки; масив `pins[]`, який містить відповідні конфігураційні дані для кожного модуля. У циклі `for` відбувається послідовне копіювання значень із масиву `pins` у відповідні поля структури `modules`. Це включає такі апаратні сигнали, як `triggerPin`, `echoPin`, `safePin`, `warningPin`, `dangerPin`, `buzzerPin` та `signalPin`. Таким чином забезпечується правильне відображення логічної структури модуля на фізичні входи та виходи мікроконтролера. Крім того, функція встановлює початкові значення для внутрішніх змінних: `lastBlink` та `lastNoise` обнуляються, що означає відсутність попередніх подій; `lightState` та `noiseState` встановлюються у нуль, що відповідає початковому стану

Цитування: 0%

id: 14

«вимкнено».

Після цього викликається функція `myPinMode` для кожного піну, що визначає його режим роботи: `OUTPUT` для керуючих сигналів та `INPUT` для сигналів з датчиків. Це забезпечує коректну взаємодію між програмним забезпеченням та апаратною частиною системи. 4.4. Функції у файлі `Interact.cpp` Файл `Interact.cpp` реалізує логіку подачі аудіо візуальної інформації. Константи (пороги відстаней та інтервали миготіння/звуку) визначаються на початку файлу. Для керування світлодіодами та бузером використовуються допоміжні функції, які оперують через `mymyDigitalWrite()` при зміні логічних рівнів виходів. Оскільки в деяких випадках світлодіоди мають спільний катод/анод і інверсну логіку, допоміжні функції інкапсулюють цю специфіку, змінюючи рівні через `mymyDigitalWrite()` відповідно до апаратної конфігурації. На початку задано константи. Такий підхід до написання коду робить його більш структурованим та адаптивнішим до змін, оскільки може знадобитися потреба збільшити інтервал між звуковими сигналами бузера, або зменшити паузу між світловими сигналами. Нижче наведено таблиця зі назвами констант та їх призначенням. Змн. Арк. № докум. Підпис Дата Арк. 23 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Таблиця 4.3 Типи констант та їх призначення в файлі `Interact.cpp` Константа Пояснення `FAR_LIMIT` Це порогове значення відстані (200 мм), яке визначає, що об'єкт знаходиться достатньо далеко й не становить небезпеки. По суті, воно позначає

Цитування: 0%

id: 15

«зону комфорту»,

де система може працювати спокійно без активних попереджень. `SAFETY_LIMIT` Відстань у 100 мм, яка сигналізує про наближення об'єкта до межі безпеки. Це вже не

Цитування: 0%

id: 16

«далеко»,

але ще й не критично близько. Використовується для активації попереджувальних сигналів, щоб користувач звернув увагу. `CLOSE_LIMIT` Значення 50 мм, яке вказує на те, що об'єкт знаходиться дуже близько. Це межа, після якої система переходить у режим

Цитування: 0%

id: 17

«майже небезпечно»,

і сигнали стають більш інтенсивними. `FAR_BLINK_INTERVAL` Інтервал миготіння світлодіода дорівнює нулю, тобто він світиться постійно. Це символізує стабільність і відсутність загрози. `SAFETY_BLINK_INTERVAL` Світлодіод блимає раз на секунду (1000 мс). Це помірна інтенсивність, який нагадує користувачу про наближення об'єкта, але не створює відчуття паніки. `CLOSE_BLINK_INTERVAL` Частота миготіння — кожні 500 мс. Це вже більш настирливий сигнал, який підкреслює, що об'єкт знаходиться небезпечно близько. Змн. Арк. № докум. Підпис Дата Арк. 24 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Константа Пояснення `DANGEROUS_BLINK_INTERVAL` Дуже швидке миготіння (200 мс), яке сприймається як сигнал тривоги. Використовується для позначення критичної небезпеки. `FAR_BUZZ_INTERVAL` Бузер подає короткі сигнали кожні 2000 мс (раз на дві секунди). Це легке нагадування, що система активна, але небезпеки немає. `SAFETY_BUZZ_INTERVAL` Інтервал у 1500 мс робить звук рідкісним і ненав'язливим, але він все одно попереджає про наближення об'єкта. `CLOSE_BUZZ_INTERVAL` Бузер сигналізує кожні 500 мс, створюючи відчуття терміновості й синхронізуючись із більш швидким миготінням світлодіода. `DANGEROUS_BUZZ_INTERVAL` Інтервал у 200 мс означає дуже часті звукові сигнали, які разом із швидким миготінням світлодіода створюють інтенсивне попередження про критично малу відстань до об'єкта. Для опису логіки роботи системи використовується перелік станів (`enum`). Це дозволяє задати набір констант, які відповідають різним рівням небезпеки залежно від відстані до об'єкта. Використання `enum` робить код більш зрозумілим, структурованим та захищає від помилок, оскільки змінна стану може приймати лише визначені значення. За визначення стану відповідає функція `getState`. Вона приймає в аргументах відстань, порівнює її із заданими лімітами та повертає відповідний тип, визначення якого базується на переданому значенні. Змн. Арк. № докум. Підпис Дата Арк. 25 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ Створені функції для переключення стану світлодіодів. На перший погляд, це може здатися недоречним, оскільки переключення стану світлодіодів здійснюється просто. Проте, в

даному пристрою використовується світлодіоди, які здатні світити трьома кольорами і мають спільний катод. Через це, нам для того що б світився певний колір, необхідно замкнути відповідну ніжку на землю. Іншими словами, необхідно змінити стан піну на [LOW](#). Це є контрінтуїтивним, тому було прийнято рішення: створити прості функції-помічники, що б синтаксис був зрозумілішим. Функція [updateBlink](#) — це

Цитування: 0%

id: 18

«керівник»

світлодіодів, який вирішує, коли його вмикає і вимикає, щоб створити ефект миготіння. Вона працює не через просту затримку ([delay](#)), яка зупиняє всю програму, а через системний таймер [millis\(\)](#), що рахує час від запуску мікроконтролера. Завдяки цьому світлодіод може блимати з потрібною частотою, а інші частини програми продовжують працювати паралельно, наприклад інші світлодіоди. Якщо інтервал миготіння дорівнює нулю, світлодіод просто горить постійно — це сигналізує про стабільний стан. Якщо інтервал заданий, то функція перевіряє, чи минув потрібний час від останнього перемикавання ([lastBlink](#)). Якщо так — вона змінює стан змінної [lightState](#) і відповідно вмикає або вимикає світлодіод. Таким чином формується ритмічне миготіння. Такий підхід називають неблокуючим керуванням, і він важливий для систем реального часу: мікроконтролер не

Цитування: 0%

id: 19

«зависає»

на очікуванні, а може одночасно обробляти сигнали від датчика відстані, керувати бузером чи виконувати інші завдання. В аргументах приймає посилання на структуру типу [Module](#), пін та інтервал миготіння. Викликають [updateBlink](#) лише допоміжні функції, які викликаються залежно від відстані. Їх буде описано нижче. [UpdateBlink\(\)](#) працює аналогічним чином. [farInteraction](#), [safetyCloseInteraction](#), [closeInteraction](#), [dangerousCloseInteraction](#) — чотири функції, які можна описати як набір

Цитування: 0%

id: 20

«сценаріїв поведінки»

системи, які визначають, що саме має робити індикація залежно від того, наскільки близько знаходиться об'єкт. Кожна функція відповідає за свій рівень небезпеки: коли об'єкт далеко — працює лише зелений Змн. Арк. № докум. Підпис Дата Арк. 26 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ світлодіод і бузер подає рідкісні сигнали; коли відстань скорочується до

Цитування: 0%

id: 21

«зони безпеки»

— зелений світлодіод починає блимати, а бузер звучить частіше; при ще більшому наближенні активується жовтий попереджувальний світлодіод із більш інтенсивним миготінням та відповідним ритмом бузера; а у випадку небезпечної близькості система переходить у режим тривоги — червоний світлодіод блимає швидко, а бузер видає майже безперервний сигнал. Таким чином, ці функції реалізують просту, але зрозумілу для користувача градацію: чим ближче об'єкт, тим сильніше й настирливіше система сигналізує. У дипломі можна подати це як приклад модульного підходу — кожна функція відповідає за окремий сценарій, але всі разом утворюють узгоджену систему світлової та звукової індикації, яка працює на основі машини станів і дозволяє легко змінювати або доповнювати поведінку системи. Функція [interact](#) є головним інтерфейсом однойменного файлу. Її завдання дуже просте, але саме в цьому полягає велика перевага: вона отримує відстань до об'єкта, визначає відповідний стан за допомогою функції [getState](#), а далі через конструктор [switch](#) викликає потрібну функцію взаємодії ([farInteraction](#), [safetyCloseInteraction](#), [closeInteraction](#) або [dangerousCloseInteraction](#)). Такий підхід робить код зрозумілим і легко підтримуваним — головна функція не містить складних умов чи запутаної логіки, а лише розподіляє роботу спеціалізованим підфункціям. Це дозволяє розділити відповідальність: кожна допоміжна функція відповідає за свій сценарій індикації, а [interact](#) виступає як

Цитування: 0%

id: 22

«керівник»,

що визначає, який сценарій має бути виконаний у даний момент. У дипломі можна підкреслити, що така структура є прикладом модульного та інтуїтивно зрозумілого програмування: головний інтерфейс простий, але саме завдяки цьому система легко розширюється, а її поведінка залишається прозорою для розробника та користувача, що захоче змінити логіку опрацювання відстані. Змн. Арк. № докум. Підпис Дата Арк. 27 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ 4.5. Скетч для [Arduino](#) В скетчі для Ардуіно було імпортовано бібліотеку

Цитування: 0%

id: 23

«[NewPing](#)».

Вона створена для полегшення та оптимізації роботи з ультразвуковими датчиками, як от той, що використовується у проєкті: [HC-SR04](#). В цій бібліотеці реалізована фільтрації

результатів вимірювання, перетворення значень датчиків на одиниці вимірювання відстані. Завдяки ній, не потрібно реалізовувати фільтрацію, вручну вимірювати час відгуку датчика та можна суттєво спростити код. Спочатку в скетчі задаємо константи: Таблиця 4.3 Типи констант та їх призначення в скетчі для [Arduino](#) Назва константи Призначення константи [MAX_DISTANCE_CM](#) Необхідно для роботи [NewPing.h](#). Задається поріг відстані, яку буде виміряно. [BETWEEN_MS](#) Пауза між опитуванням датчиків. Потрібна для синхронізації опитування, щоб один датчик отримував ультразвукові сигнали від іншого. [MEDIAN_ITERS](#) Призначена для фільтрації. Задає кількість опитувань датчика, перед тим як повернути відфільтрований результат. У цьому фрагменті коду реалізовано багатомодульну систему для вимірювання відстані за допомогою ультразвукових сенсорів. Кожен модуль описується структурою [ModulePins](#), де визначені виводи для запуску імпульсу, прийому сигналу, світлової індикації рівня безпеки, підключення бузера та додаткового сигналу, що дозволяє динамічно визначати кількість [slave](#)-модулів. Це дозволяє гнучко налаштовувати конфігурацію кожного модуля залежно від апаратної схеми. У функції [setup\(\)](#) програма визначає кількість активних модулів, заповнює масив структур і створює об'єкти бібліотеки [NewPing](#) для кожного сенсора. Бібліотека використовується для роботи з ультразвуковими датчиками, забезпечуючи точні вимірювання завдяки усередненню кількох результатів. Функція [getFilteredDistance\(\)](#) виконує вимірювання відстані, використовуючи метод [ping_median\(\)](#), який зменшує похибку за рахунок статистичної обробки. Отриманий час проходження сигналу переводиться у сантиметри. Змн. Арк. № докум. Підпис Дата Арк. 28 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ У циклі [loop\(\)](#) програма послідовно опитує всі модулі, отримує відстань і викликає функцію [interact\(\)](#), яка відповідає за реакцію системи. Залежно від результату можуть активуватися світлодіоди різних кольорів та/або звуковий сигналізатор. Таким чином реалізується багаторівнева система попередження про небезпеку. Функції обгортки [myDigitalWrite\(\)](#) [myPinMode\(\)](#) та [myDigitalRead\(\)](#) призначені для коректної роботи з розширювачем портів. Їх логіка проста: якщо номер піна менший від 16, значить що він знаходиться на розширювачі, а не [Arduino](#), інакше на самій платі. Змн. Арк. № докум. Підпис Дата Арк. 29 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ РОЗДІЛ 5. ТЕСТУВАННЯ ТА УСУНЕННЯ ПОМИЛОК 5.1. Виявлення та виправлення апаратних недоліків Під час самої збірки було допущено деякі помилки. Найбільш критичною стало неправильне обтискання роз'єму [Molex](#). Незважаючи на те, що проводи були обтиснуті правильно, вони були поміщені не в той роз'єм. Якщо пояснити детальніше, є два типу роз'ємів [Molex](#). [Male](#) та [Female](#). В кожному з них є 8, чи 4 [Dupont](#) пінів, в залежності від роз'єму. Для правильного обтискання [Molex](#), потрібно обтиснути провід роз'ємом [Dupont male](#) та вставити у [Molex female](#), а [Dupont female](#) у [Molex male](#). На мій погляд, це є контрінтуїтивним, оскільки у нас з'являється роз'єм [Molex male](#), а всередині у нього [Dupont female](#). Це призвело до моєї помилки. Тому, я був вимушений переробляти роз'єми. Дуже великим недоліком, окрім вище описаного, також є

Цитування: 0%

id: 24

«одноразовість»

такого рішення. Дістати провід з них дуже складно та їх неможливо знайти в магазинах мого міста. Резюмуючи, в роз'ємах багато недоліків, та я не маю змогу їх знайти в магазині. Через це, було прийнято рішення замінити їх на інші. Нові роз'єми мають лише один недолік — вони значно більші, проте, враховуючи що це макет системи виявлення перешкод, а ще не повноцінний продукт, то цим можна знехтувати. 5.2. Виявлення та виправлення недоліків у коді Код було створено вже достатньо надійним та продуманим, що призвело до мінімальної кількості виправлення помилок у ньому. Під час компіляції було виявлено, що бібліотека [NewPing.h](#) теж використовує функцію [tone\(\)](#), як і бузери що призвело до помилки. Тому було прийнято рішення, що бузери будуть використовувати лише [myDigitalWrite\(buzzerPin, HIGH\)](#) для виконання своїх функцій. Змн. Арк. № докум. Підпис Дата Арк. 30 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. РОЗДІЛ 5. ТЕСТУВАННЯ ТА УСУНЕННЯ ПОМИЛОК Літ. Акрушів 2 ЛНУ Кафедра ІТС, Гр.4КІ . 5.3. Ідеї для покращення після проведеного тестування Провівши тестування, було виявлено деякі недоліки та ідеї для покращення. Перш за все, необхідно покращити кріплення для настінного кріплення кожного з модулів, оскільки в більшості випадків аналогічні пристрої кріплять на стіни скотчем, проте в нашому випадку, через використання проводу, що робить конструкцію важчою, доцільно реалізувати кріплення на стіну. Принаймні зробити отвір під гвіздок. Оскільки споживання приладу невисоке, в майбутньому можливо буде обдумати використання батарейок для його живлення. Необхідно буде замінити світлодіод на яскравіший, або навіть використовувати [ws2812](#) для покращених світлових ефектів. Хоча для демонстраційних цілей та використання в темних приміщеннях, цього світлодіоду достатньо. Для більшої зручності використання, в майбутньому можна буде замінити роз'єми на практичніші, хоча це призведе до збільшення вартості. Змн. Арк. № докум. Підпис Дата Арк. 31 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ ВИСНОВКИ У цьому дипломному проєкті ми крок за кроком вибудували цілісну концепцію системи допомоги при паркуванні. Спершу, у теоретичній частині, ми розібралися з основними

Цитування: 0%

id: 25

«цеглинками»

майбутнього прототипу — ультразвуковими датчиками, [Arduino](#), світлодіодами та бузером. Саме вони формують фундамент системи, дозволяючи вимірювати відстань, подавати світлові й звукові сигнали та забезпечувати взаємодію з водієм. Далі ми звернулися до огляду існуючих технологій і рішень. Тут стало очевидно: ультразвукові сенсори — найбільш надійний вибір для гаражних умов. Інфрачервоні та оптичні аналоги виявилися занадто чутливими до світла, кольору кузова чи пилу. Аналіз комерційних пристроїв показав, що ринок пропонує прості, але закриті системи, які не дають простору для модифікацій. Це лише підтвердило потребу у відкритій модульній платформі. У третій частині ми перейшли до

Цитування: 0%

id: 26

«заліза»

— апаратної складової. Було обґрунтовано вибір компонентів, побудовано схему підключення та показано, що використання [Arduino](#) робить прототип доступним і гнучким. Така архітектура дозволяє легко масштабувати систему й адаптувати її під різні умови. Четвертий розділ присвячено програмному забезпеченню. Ми описали структуру модулів, функції та створили скетч для [Arduino](#). Програмна частина забезпечує точність вимірювань завдяки алгоритмам фільтрації та почерговому опитуванню сенсорів, що дозволяє уникнути перешкод і підвищити стабільність роботи. П'ятий розділ став своєрідною

Цитування: 0%

id: 27

«репетицію»

перед виходом у реальність. Під час тестування ми виявили й усунули недоліки як у апаратній, так і в програмній частині. Система після корекцій показала стабільну роботу, а додатково були сформульовані ідеї для майбутніх удосконалень. Змн. Арк. № докум. Підпис Дата Арк. 32 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. ВИСНОВКИ Літ. Акрушів 2 ЛНУ Кафедра ІТС, Гр.4КІ . У підсумку вся робота довела: створення стаціонарної модульної системи допомоги при паркуванні — цілком реальне завдання. Вона поєднує простоту апаратної реалізації з гнучкістю програмного забезпечення, залишається доступною для широкого кола користувачів і має потенціал для розвитку. Це не просто прототип, а платформа, яка може стати основою для нових рішень у сфері безпечного та комфортного паркування. Змн. Арк. № докум. Підпис Дата Арк. 33 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ Змн. Арк. № докум. Підпис Дата Арк. 34 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ Літ. Акрушів 2 ЛНУ Кафедра ІТС, Гр.4КІ . 1. [Wong J.](#) Використання ультразвукового сенсора: посібник.

[bjultrasonic](#). URL: <https://www.bjultrasonic.com/uk/how-to-use-ultrasonic-sensor/> 2. Типи та застосування звукових сигналізаторів | [Transfer Multisort Elektronik Ukraine](#). TME. URL: <https://www.tme.eu/ua/news/library-articles/page/63624/tipi-ta-zastosuvannia-zvukovikh-signalizatoriv/> (дата звернення: 29.04.2026) 3. [VisitUkraine Today](#). Автомобільний парк України старіє: де у 2025 році продаватимуться наймолодші та найстаріші автомобілі. [Visit Ukraine - SERVICE PORTAL ABOUT UKRAINE](#) 24/7. URL: <https://visitukraine.today/uk/blog/7503/the-ukrainian-car-fleet-is-aging-where-the-youngest-and-oldest-cars-will-> 2025 4. Середній вік вантажівок в Україні – 18 років. Як це змінити?. Інститут досліджень авторинку • Аналітика та прогнозування. URL: <https://eauto.org.ua/news/376-seredniy-vik-vantazhivok-v-ukrajini-18-rokiv-yak-ce-zminiti> 5. [Adjustable Garage Side Parking Sensor — STKR Concepts](#) [Electronic Resource]. — Mode of access: <https://stkrconcepts.com/products/adjustable-garage-side-parking-sensor> 6. [Zone Tech Car Parking Sensor on Amazon](#) [Electronic Resource]. — Mode of access: <https://www.amazon.com/> 7. Офіційний каталог виробника: [LiftMaster Commercial Photo Eyes](#) [Electronic Resource]. — Режим доступу: <https://www.liftmaster.com/cps-u-commercial-photo-eyes/p/CPS-U> 8. Інженерний опис та магазин компонентів: [Banana Robotics](#). [Sharp GP2Y0A21YK0F IR Distance Sensor](#) [Electronic Resource]. — Режим доступу: <https://www.bananarobotics.com/shop/Sharp-GP2Y0A21YK0F-IR-Distance-Sensor> 9. Офіційний магазин розробника: [MAXSA Innovations](#). [Park Right Dual Garage Laser Park](#) [Electronic Resource]. — Режим доступу: <https://www.maxsainnovations.com/park-right-dual-garage-laser-park/> 10. Профільна технічна стаття:

Цитування: 0,01%

id: 28

«Вимірювання відстані: лазерний чи ультразвуковий датчик?»

<https://peko.com.ua/statti/laser-or-ultrasonic> 11. Документація [Arduino.ua](#) URL: <https://docs.arduino.cc/language-reference/en/functions/time/millis/> Змн. Арк. № докум. Підпис Дата Арк. 35 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ ДОДАТОК А Змн. Арк. № докум. Підпис Дата Арк. 36 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. ДОДАТОК А Літ. Акрушів 7 ЛНУ Кафедра ІТС, Гр.4КІ . Лістинг А.1 – Код

Цитування: 0%

id: 29

«IC.100»

```
#include Arduino.h #include NewPing.h #include Wire.h #include Adafruit_MCP23017.h #include
```

Цитування: 0%

id: 30

```
"module.h"
```

```
#define MAX_DISTANCE_CM 200 #define BETWEEN_MS 45 #define MEDIAN_ITERS 3
Adafruit_MCP23017 mcp; // 2C expander // модулі ModulePins module1 = { .triggerPin = 2,
.echoPin = 3, .safePin = 0, .warningPin = 3, .dangerPin = 6, .buzzerPin = 7, .signalPin = 8 };
ModulePins module2 = { .triggerPin = 9, .echoPin = 10, .safePin = 9, .warningPin = 12,
.dangerPin = 15, .buzzerPin = A0, .signalPin = A1 }; ModulePins module3 = { .triggerPin = A2,
.echoPin = A3, .safePin = 1, .warningPin = 4, .dangerPin = 7, .buzzerPin = -1, .signalPin = 0 };
ModulePins pins[] = { module1, module2, module3 }; Змн. Арк. № докум. Підпис Дата Арк. 37
ITC.4KI.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ size_t n = sizeof(pins) /
sizeof(pins[0]); int count; Module modules[3]; NewPing* sonars[3]; // обгортки void
myPinMode(int pin, int mode) { if (pin = 0 && pin 16) mcp.pinMode(pin, mode); else pinMode(pin,
mode); } void myDigitalWrite(int pin, int value) { if (pin = 0 && pin 16) mcp.digitalWrite(pin,
value); else digitalWrite(pin, value); } int myDigitalRead(int pin) { if (pin = 0 && pin 16) return
mcp.digitalRead(pin); else return digitalRead(pin); } void setup() { Serial.begin(9600);
mcp.begin(); // init MCP count = findModulesCount(pins, n); fillModules(modules, count, pins); for
(int i = 0; i count; i++) { sonars[i] = new NewPing(modules[i].triggerPin, modules[i].echoPin,
MAX_DISTANCE_CM); } } long getFilteredDistance(NewPing* sonar) { unsigned int uS = sonar-
ping_median(MEDIAN_ITERS); return uS / US_ROUNDTRIP_CM; } void loop() { for (int i = 0; i
count; i++) { long distance = getFilteredDistance(sonars[i]); interact(&modules[i], distance); }
delay(BETWEEN_MS); } кінець лістингу A.1 Лістинг A.2 – Код
```

Цитування: 0%

id: 31

```
<<"module.h">>
```

Змн. Арк. № докум. Підпис Дата Арк. 38 ITC.4KI.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА
ДОКУМЕНТУ #ifndef MODULE_H #define MODULE_H #include Arduino.h #ifdef __cplusplus

Цитування: 0%

id: 32

```
" "
```

```
{ #endif typedef struct { int triggerPin; int echoPin; int safePin; int warningPin; int dangerPin; int
buzzerPin; int signalPin; int lightState; int noiseState; unsigned long lastBlink; unsigned long
lastNoise; } Module; typedef struct { int triggerPin; int echoPin; int safePin; int warningPin; int
dangerPin; int buzzerPin; int signalPin; } ModulePins; int findModulesCount(const ModulePins
pins[], int size); void fillModules(Module modules[], int count, const ModulePins pins[]); void
interact(Module *module, long distance); #ifdef __cplusplus } #endif #endif кінець лістингу A.2
Лістинг A.3 – Код
```

Цитування: 0%

id: 33

```
<<"module.c">>
```

Змн. Арк. № докум. Підпис Дата Арк. 39 ITC.4KI.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА
ДОКУМЕНТУ #include

Цитування: 0%

id: 34

```
"module.h"
```

```
#include Arduino.h // кількість модулів int findModulesCount(const ModulePins pins[], int size) {
int count = 0; for (int i = 0; i size; i++) { myPinMode(pins[i].signalPin, INPUT); if
(!myDigitalRead(pins[i].signalPin)) { count++; } } return count; } // заповнення void
fillModules(Module modules[], int count, const ModulePins pins[]) { for (int i = 0; i count; i++) {
modules[i].triggerPin = pins[i].triggerPin; modules[i].echoPin = pins[i].echoPin;
modules[i].safePin = pins[i].safePin; modules[i].warningPin = pins[i].warningPin;
modules[i].dangerPin = pins[i].dangerPin; modules[i].buzzerPin = pins[i].buzzerPin;
modules[i].signalPin = pins[i].signalPin; modules[i].lastBlink = 0; modules[i].lastNoise = 0;
modules[i].lightState = 0; modules[i].noiseState = 0; myPinMode(modules[i].triggerPin,
OUTPUT); myPinMode(modules[i].echoPin, INPUT); myPinMode(modules[i].safePin, OUTPUT);
myPinMode(modules[i].warningPin, OUTPUT); myPinMode(modules[i].dangerPin, OUTPUT);
myPinMode(modules[i].buzzerPin, OUTPUT); myPinMode(modules[i].signalPin, INPUT); } } кінець
лістингу A.3 Лістинг A.4 – Код
```

Цитування: 0%

id: 35

```
<<"interact.cpp">>
```

Змн. Арк. № докум. Підпис Дата Арк. 40 ITC.4KI.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА
ДОКУМЕНТУ #include

Цитування: 0%

id: 36

```
"module.h"
```

```
#include Arduino.h #define FAR_LIMIT 200 #define SAFETY_LIMIT 100 #define CLOSE_LIMIT 50
#define FAR_BLINK_INTERVAL 0 #define SAFETY_BLINK_INTERVAL 1000 #define
CLOSE_BLINK_INTERVAL 500 #define DANGEROUS_BLINK_INTERVAL 200 #define
FAR_BUZZ_INTERVAL 2000 #define SAFETY_BUZZ_INTERVAL 1500 #define
```

```

CLOSE_BUZZ_INTERVAL 500 #define DANGEROUS_BUZZ_INTERVAL 200 typedef enum { Far,
SafetyClose, Close, DangerousClose } States // LED (анод) void ledOn(int pin) {
myDigitalWrite(pin, LOW); } void ledOff(int pin) { myDigitalWrite(pin, HIGH); } States
getState(long distance) { if (distance FAR_LIMIT) return Far; else if (distance SAFETY_LIMIT) return
SafetyClose; else if (distance CLOSE_LIMIT) return Close; else return DangerousClose; } //
миготіння static void updateBlink(Module *m, int pin, unsigned long interval) { if (interval == 0)
{ ledOn(pin); m- lightState = 1; return; } unsigned long now = millis(); Змн. Арк. № докум.
Підпис Дата Арк. 41 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА ДОКУМЕНТУ if (now - m-
lastBlink = interval) { m- lastBlink = now; m- lightState = !m- lightState; if (m- lightState)
ledOn(pin); else ledOff(pin); } } // бузер static void updateBuzzer(Module *m, int buzzerPin,
unsigned long interval) { if (buzzerPin 0) return; if (interval == 0) { myDigitalWrite(buzzerPin,
LOW); m- noiseState = 0; return; } unsigned long now = millis(); if (now - m- lastNoise = interval)
{ m- lastNoise = now; m- noiseState = !m- noiseState; myDigitalWrite(buzzerPin, m- noiseState ?
HIGH : LOW); } } // стани void farInteraction(Module *m) { ledOff(m- warningPin); ledOff(m-
dangerPin); updateBlink(m, m- safePin, FAR_BLINK_INTERVAL); updateBuzzer(m, m- buzzerPin,
FAR_BUZZ_INTERVAL); } void safetyCloseInteraction(Module *m) { ledOff(m- warningPin);
ledOff(m- dangerPin); updateBlink(m, m- safePin, SAFETY_BLINK_INTERVAL); updateBuzzer(m, m-
buzzerPin, SAFETY_BUZZ_INTERVAL); } void closeInteraction(Module *m) { ledOff(m- safePin);
ledOff(m- dangerPin); updateBlink(m, m- warningPin, CLOSE_BLINK_INTERVAL); updateBuzzer(m,
m- buzzerPin, CLOSE_BUZZ_INTERVAL); } void dangerousCloseInteraction(Module *m) { Змн.
Арк. № докум. Підпис Дата Арк. 42 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ НАЗВА
ДОКУМЕНТУ ledOff(m- safePin); ledOff(m- warningPin); updateBlink(m, m- dangerPin,
DANGEROUS_BLINK_INTERVAL); updateBuzzer(m, m- buzzerPin, DANGEROUS_BUZZ_INTERVAL);
} // головна void interact(Module *module, long distance) { States state = getState(distance);
switch (state) { case Far: farInteraction(module); break; case SafetyClose:
safetyCloseInteraction(module); break; case Close: closeInteraction(module); break; case
DangerousClose: dangerousCloseInteraction(module); break; } } кінець лістингу А.4 ДОДАТОК Б
Змн. Арк. № докум. Підпис Дата Арк. 43 ІТС.4КІ.0126.03-ПЗ Розроб. Бородавка Р.В. Керівник
Донченко В.Ю. Реценз. Козуб Ю.Г. Н. Контр. Зав. каф. Семенов М.А. ДОДАТОК Б Літ. Акрушів
2 ЛНУ Кафедра ІТС, Гр.4КІ . Рис. Б.1. Підлеглий модуль до оновлення. Рис. Б.2. Корпус
майстер модуля до заміни роз'ємів. Рис. Б.3. Підлеглий модуль з покращеними роз'ємами
та пайкою. Змн. Арк. № докум. Підпис Дата Арк. 44 ІТС.4КІ.0126.03-ПЗ НАЗВА ДОКУМЕНТУ
НАЗВА ДОКУМЕНТУ Рис. Б.4. Майстер модуль з новими роз'ємами.

```

Заявление об ограничении ответственности:

Этот отчет должен быть правильно истолкован и проанализирован квалифицированным специалистом, который несет ответственность за оценку!

Любая информация, представленная в этом отчете, не является окончательной и подлежит ручному просмотру и анализу. Пожалуйста, следуйте инструкциям: [Рекомендации по оценке](#)

Детектор Плагиата - Ваше право на оригинальность! ☐ SkyLine LLC