

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ
ТА БЕЗПЕКИ**

Кафедра інформаційних технологій

**РОЗРОБКА ГОДИННИКА ЗІ ЗВУКОВИМ ОПОВІЩЕННЯМ З
ВИКОРИСТАННЯМ ПЛАТФОРМИ ARDUINO ТА MP3 МОДУЛЯ
DFPLAYER**

кваліфікаційна робота

здобувача вищої освіти

4 курсу денної форми навчання

Петра РУБАЯ

Науковий керівник:

доцент, кандидат технічних наук

Ігор ФАРМАГА

Рецензент:

вчене звання, науковий ступінь

(Ім'я ПРІЗВИЩЕ рецензента)

Кваліфікаційна робота допущена до захисту

«___» _____ 2025 р., протокол № _____

Завідувач кафедри інформаційних технологій

_____ Ігор ФЕДЧАК

(підпис)

Львів
2025

АНОТАЦІЯ

РУБАЙ П. Розробка годинника зі звуковим оповіщенням на платформі Arduino і MP3 модулі DFPlayer. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». – Львівський державний університет внутрішніх справ, МВС України, Львів, 2025.

У роботі представлено годинник зі звуковим оповіщенням на базі Arduino та MP3-модуля DFPlayer. Проведено огляд технічних рішень, обґрунтовано вибір компонентів, розроблено схему та алгоритм роботи пристрою. Тестування підтвердило працездатність системи. У висновках запропоновано шляхи її удосконалення та можливості застосування у сфері безпеки.

Ключові слова: Arduino, DFPlayer Mini, звукове сповіщення, RTC DS3231, годинник зі звуковим оповіщенням.

ABSTRACT

Rubai P. Development of a clock with sound alert on the Arduino platform and MP3 modules DFPlayer.

Research for obtaining a bachelor's degree in specialty 262 “Law enforcement information systems and technologies.” - Lviv State University of Internal Affairs, Ministry of Internal Affairs of Ukraine, Lviv, 2025.

The paper presents a watch with sound alert based on Arduino and the DFPlayer MP3 module. A review of technical solutions was conducted, the choice of components was substantiated, and a circuit and algorithm for the device were developed. Testing has confirmed the system's performance. In conclusion, the ways of its improvement and possibilities of application in the field of security are proposed.

Arduino, DFPlayer Mini, sound alert, RTC DS3231, clock with sound alert.

Зміст

АНОТАЦІЯ.....	2
ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ АУДІОІНТЕРФЕЙСУ В ГОДИННИКАХ.....	8
1.1. Комерційні пристрої зі звуковими сигналами.....	8
1.2. Відкриті проекти на платформі Arduino.....	11
1.3. Порівняльний аналіз комерційних та відкритих рішень.....	12
Висновок розділу 1.....	14
РОЗДІЛ 2 ОПИС АПАРАТНОЇ ЧАСТИНИ.....	16
2.1. Обґрунтування вибору технічних засобів для створення інформаційної системи.....	16
2.2. Arduino Uno / Nano.....	17
2.3. LCD-дисплеї (I2C).....	19
2.4. TMP36 – сенсор температури.....	20
2.5. Кнопковий інтерфейс.....	21
2.6. DFPlayer Mini – MP3-модуль.....	22
2.7. Динамік.....	23
2.8. Схема підключення.....	24
Висновок розділу 2.....	27
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	28
3.1. Структура програмного забезпечення.....	28
3.2. Реалізація меню пристрою.....	29
3.3. Обробка кнопок.....	32
3.4. Робота з TMP36.....	34
3.5. Взаємодія з DFPlayer Mini.....	35

	4
3.6. Відлагодження та тестування коду.....	37
Висновок розділ 3.....	38
РОЗДІЛ 4 ТЕСТУВАННЯ.....	39
4.1. Методика тестування.....	39
4.2. Результати тестування.....	40
4.3. Аналіз результатів тестування.....	41
Висновок розділу 3.....	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТКИ.....	48
Додаток А.....	49

ВСТУП

Сучасні технології відкривають нові можливості для створення пристроїв, що підвищують якість життя людей з особливими потребами. Особливо актуальною є розробка пристроїв для осіб із порушеннями зору, які потребують альтернативних способів сприйняття інформації. Одним із таких рішень є годинник зі звуковим оповіщенням, що дозволяє людям з вадами зору орієнтуватися в часі без необхідності зорового контакту з дисплеєм.

Розробка багатофункціонального годинника зі звуковим оповіщенням також має безпосереднє відношення до спеціальності «Правоохоронні інформаційні системи та технології», оскільки проєкт демонструє інтеграцію програмно-апаратних засобів для вирішення завдань соціальної адаптації осіб з вадами зору, включаючи військовослужбовців, що постраждали внаслідок бойових дій.

Особливу увагу в цьому контексті варто звернути на **потреби військовослужбовців, які втратили зір внаслідок бойових дій**. В умовах сучасних збройних конфліктів, зокрема під час повномасштабної війни в Україні, багато військових зазнали тяжких поранень, у тому числі втратили зір. Такі травми істотно змінюють спосіб життя людини, вимагаючи створення нових інструментів для її адаптації в цивільному середовищі.

Прикладом може слугувати ситуація, коли колишній борець, який не бачить, потребує знання часу для прийому ліків, дотримання режиму або виконання побутових справ. У таких випадках використання звичайного годинника неможливе без сторонньої допомоги. Саме тут і стає незамінним годинник зі **звуковим оповіщенням**, що дозволяє людині самостійно дізнатися поточний час, активувати таймер чи будильник.

Окрім того, такий пристрій може бути використаний у **реабілітаційних центрах** або як індивідуальний засіб допомоги в побуті. Гнучкість програмної частини дозволяє адаптувати систему під потреби кожного користувача, додаючи нові функції або змінюючи мову й стиль озвучення. Завдяки

використанню доступних і недорогих компонентів, пристрій можна легко відтворити або доопрацювати навіть у домашніх умовах.

Метою кваліфікаційної роботи є розробка багатофункціонального годинника зі звуковим оповіщенням, реалізованого на основі платформи Arduino з використанням MP3 модуля DFPlayer. Пристрій передбачає використання LCD-дисплея, кнопочового інтерфейсу, сенсора температури та програмної логіки, що забезпечує роботу в кількох режимах: відображення часу, будильник, таймер, мелодійне оповіщення та вимірювання температури.

Об'єктом дослідження є процес взаємодії користувача з мікроконтролерною системою оповіщення про події з використанням аудіовиходу.

Практична цінність роботи полягає у створенні доступного, відкритого й адаптивного пристрою, який може бути:

- використаний людьми з вадами зору;
- застосований у реабілітаційних програмах для військових;
- впроваджений у соціальних або навчальних закладах;
- використаний як навчальний проєкт для вивчення принципів мікроконтролерної розробки.

Для досягнення поставленої мети в межах кваліфікаційної роботи було визначено такі основні завдання:

Провести аналіз існуючих аналогів та технічних рішень, що виконують подібні функції.

Було здійснено огляд пристроїв з аудіооповіщенням, розумних годинників та систем допомоги для людей з вадами зору. Розглянуто як комерційні продукти, так і відкриті проєкти на основі Arduino. Визначено основні недоліки, які можна усунути в межах даного проєкту.

Вибрати оптимальну апаратну платформу, набір електронних компонентів та модулі для реалізації поставленої задачі. На основі порівняльного аналізу було обрано Arduino як центральний контролер, LCD-дисплеї з I2C-інтерфейсом для візуального виведення

інформації, температурний сенсор TMP36, кнопки керування та звуковий модуль DFPlayer Mini для голосового супроводу.

Розробити схему з'єднання компонентів та створити апаратну частину пристрою.

Сформовано електричну схему з використанням середовища Tinkercad, що дозволяє швидко протестувати та візуалізувати підключення. Підібрано номінали елементів, визначено логіку керування кнопками та підключення живлення.

Реалізувати програмне забезпечення для забезпечення функціонування всіх режимів пристрою.

Написано програмний код у середовищі Arduino IDE, який забезпечує перемикання режимів (годинник, таймер, будильник, температура, мелодії), обробку натискань кнопок, озвучення повідомлень за допомогою DFPlayer, а також обчислення температури з аналогового датчика.

Провести тестування пристрою та аналіз його роботи в реальних умовах.

Перевірено функціональність усіх режимів, зокрема точність вимірювання температури, чіткість звукового сигналу, стабільність роботи кнопок та дисплеїв. Оцінено ефективність запропонованого рішення та обґрунтовано можливість його практичного застосування.

У сучасному світі ринок пристроїв зі звуковими сигналами для людей з вадами зору представлений широким спектром рішень. Серед них – класичні «говорильні» годинники, смарт-годинники з голосовими повідомленнями, а також спеціалізовані аудіопристрої для орієнтації в просторі. Основними прикладами таких пристроїв є:

[illegible]

Схема говорильного годинника заснована на мікроконтролері PIC16F876A, який керує всіма компонентами пристрою. Для виведення

інформації використовується LCD-дисплей, а управління функціями здійснюється за допомогою кнопок MENU, TALK, INCREASE та DECREASE. Аудіофайли записані на SD-карті та відтворюються через аудіовихід, що дозволяє озвучувати поточний час і повідомлення. Система працює від стабільного живлення 5 В і 3.3 В, а стабільність роботи забезпечується осцилятором на 8 МГц. Резистивні дільники напруги та захисні діоди використовуються для узгодження сигналів між компонентами та захисту від перенапруги.

Смарт-годинники – це багатофункціональні пристрої, які поєднують функції годинника та цифрових технологій. Вони оснащені голосовими помічниками, такими як Siri, Google Assistant та Alexa, що дозволяє озвучувати час, повідомлення та нагадування. Такі пристрої зручні для людей із порушеннями зору, адже забезпечують голосове інформування про події та поточний час. Водночас, складний інтерфейс та велика кількість сенсорних елементів можуть стати проблемою для користувачів із повною втратою зору, які не мають досвіду роботи з подібними гаджетами. Крім того, деякі функції смарт-годинників вимагають зорової взаємодії з екраном, що ускладнює їх використання без сторонньої допомоги.

Смарт-годинники також підтримують будильники, таймери та синхронізацію зі смартфонами для отримання повідомлень. Популярні моделі, як Apple Watch Series 8, Samsung Galaxy Watch 6 та Amazfit Bip U Pro, забезпечують доступ до різноманітних функцій, включаючи моніторинг фізичних показників та голосове керування. Однак їхня висока вартість та потреба у підключенні до смартфона можуть бути суттєвими недоліками для деяких користувачів. Незважаючи на це, смарт-годинники залишаються зручними інструментами для тих, хто прагне отримувати інформацію у будь-який момент, не дістаючи смартфон.

Аудіопристрої з підтримкою MP3 – це годинники, що використовують MP3-модулі для відтворення заздалегідь записаних звукових файлів. Вони дозволяють адаптувати голосові повідомлення під конкретні потреби

користувача, що є особливо корисним для людей із порушеннями зору. Такий підхід забезпечує не лише озвучення поточного часу, але й відтворення будильників, таймерів та інших нагадувань у формі мелодій чи голосових повідомлень. Водночас, для людей з вадами зору такі пристрої можуть бути більш доступними, адже не потребують взаємодії з екраном.

Годинники з MP3-модулями надають можливість налаштовувати контент, додаючи у пам'ять пристрою будь-які аудіофайли – від простих голосових повідомлень до музичних композицій. Це дозволяє створити персоналізовані сповіщення, які можна використовувати як будильник чи таймер. Прикладом таких пристроїв є моделі з DFPlayer Mini Рисунок 2, які здатні відтворювати звукові файли з SD-карти за визначеним сценарієм.

Попри гнучкість у налаштуванні контенту, аудіопристрої з MP3-модулями мають певні недоліки. Вони можуть потребувати додаткової пам'яті для зберігання файлів, а також складних налаштувань для коректного відтворення повідомлень у потрібний час. Крім того, якість звучання може залежати від використовуваних динаміків, а зміна контенту вимагає перезапису файлів на SD-карті. Водночас, у порівнянні зі смарт-годинниками, такі пристрої є простішими у використанні, мають нижчу вартість та не потребують підключення до смартфона, що робить їх доступнішим рішенням для людей із вадами зору.

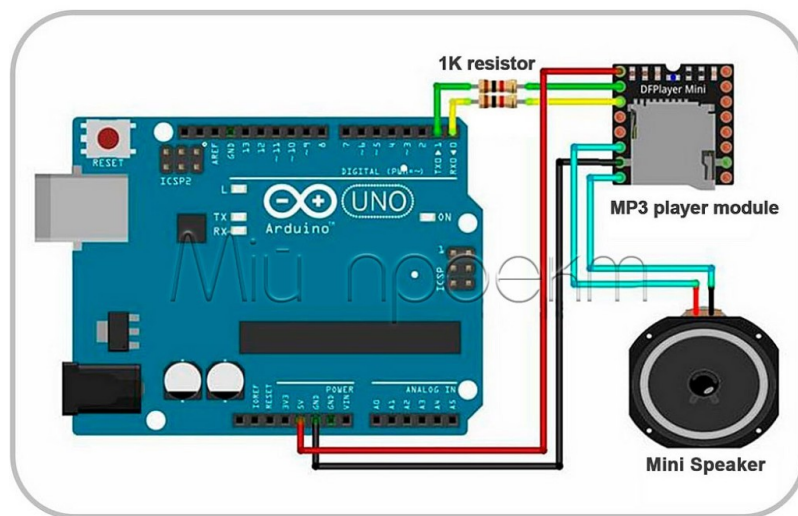


Рисунок 2 MP3-модуль схема Arduino

Основним недоліком комерційних пристроїв є висока вартість, складність ремонту та обмежені можливості адаптації під потреби конкретного користувача. Водночас, відкриті проєкти на платформі Arduino дозволяють створювати аналогічні системи з використанням доступних та дешевих компонентів, що розглянуто у наступному підрозділі.

1.2. Відкриті проєкти на платформі Arduino

Відкриті проєкти на платформі Arduino пропонують гнучкі можливості для створення пристроїв зі звуковим оповіщенням, використовуючи доступні компоненти та модулі. Завдяки відкритій архітектурі можна легко адаптувати систему під індивідуальні потреби, розширювати її функціонал та вносити зміни без значних фінансових витрат.

Arduino – це універсальна платформа з відкритим кодом, яка забезпечує інтеграцію різноманітних компонентів для створення електронних пристроїв.

Оснoву складають мікроконтролери серії ATmega, які дозволяють підключати до плати як прості кнопки та світлодіоди, так і складні модулі, такі як MP3-плеєри, LCD-дисплеї та динаміки. Це дає можливість реалізовувати системи звукового оповіщення з функцією відтворення повідомлень та мелодій, що особливо актуально для пристроїв для людей із вадами зору.

MP3-модулі, такі як DFPlayer Mini, забезпечують просту інтеграцію аудіофайлів із microSD-карти, що дозволяє реалізовувати голосові повідомлення та мелодії без складної програмної обробки. Водночас LCD-дисплеї надають можливість дублювати або доповнювати звукову інформацію текстовими повідомленнями.

Таким чином, Arduino виступає базовою платформою для створення пристроїв зі звуковим інтерфейсом, надаючи можливість адаптувати систему під конкретні потреби та налаштовувати її функціональність за допомогою доступних компонентів.

Відкриті проекти на базі Arduino орієнтовані на можливість модифікації та розширення функціоналу пристроїв. Так, до стандартних звукових інтерфейсів можна додати функції будильника, таймера, звукових нагадувань або навіть датчики для вимірювання температури. Таким чином, користувачі можуть створити унікальний пристрій, адаптований під конкретні завдання, не обмежуючись готовими рішеннями.

1.3. Порівняльний аналіз комерційних та відкритих рішень

Порівняння комерційних та відкритих рішень для реалізації аудіоінтерфейсу в годинниках дозволяє чітко визначити їх сильні та слабкі сторони, а також вибрати оптимальний підхід для конкретного проєкту. Комерційні рішення зазвичай пропонують готові пристрої зі стандартизованими функціями, тоді як відкриті проєкти на базі Arduino чи інших платформ з відкритим кодом забезпечують більшу гнучкість у налаштуванні та адаптації. У таблиці нижче представлено основні аспекти для порівняння:

Таблиця 1 Порівняльна таблиця

Аспект	Комерційні пристрої	Відкриті проєкти на Arduino
Вартість	Висока	Низька
Можливість адаптації	Обмежена	Висока
Функціональність	Вбудовані функції, що залежать від виробника	Залежить від програмної реалізації користувача
Ремонтопридатність	Складний ремонт	Легкий доступ до компонентів
Складність реалізації	Простий у використанні	Вимагає навичок програмування та пайки

Комерційні пристрої відрізняються високою якістю апаратної частини, надійністю та широким набором готових функцій. Вони можуть включати вбудовані MP3-модулі, голосових помічників, таймери та будильники. Однак їх недоліком є обмеженість у можливості зміни функціоналу та значно вища вартість. Водночас відкриті рішення дозволяють не лише інтегрувати MP3-модулі, але й створювати індивідуальні аудіоінтерфейси з можливістю запису власних голосових повідомлень та мелодій. Вони є економічно вигідними, але потребують базових знань електроніки та програмування для успішної реалізації.

Таким чином, у таблиці представлено основні аспекти для порівняння комерційних та відкритих рішень для реалізації аудіоінтерфейсу в годинниках, включаючи вартість, функціональність, можливості налаштування та рівень складності реалізації.

Висновок розділу 1

У розділі було проведено всебічний аналіз технологій для реалізації аудіоінтерфейсу в годинниках, орієнтованих на людей з вадами зору. Було розглянуто як комерційні рішення, так і відкриті проєкти на базі Arduino. З'ясовано, що комерційні пристрої, хоча й забезпечують певний базовий набір функцій оповіщення, мають високу вартість, обмежені можливості кастомізації та складність ремонту. Водночас, пристрої на основі Arduino демонструють значну гнучкість у реалізації функціональних можливостей, можливість інтеграції додаткових модулів, таких як MP3-плеєри (DFPlayer Mini) та температурні сенсори (TMP36).

Також було встановлено, що відкриті проєкти на Arduino дозволяють реалізувати не лише основні функції оповіщення, але й додати нові режими, наприклад, будильник, таймер, вимірювання температури та мелодійні сповіщення. Це робить такі системи більш привабливими для користувачів з індивідуальними потребами, зокрема для осіб, які втратили зір внаслідок бойових дій.

Порівняння комерційних і відкритих рішень продемонструвало, що проєкти на базі Arduino є більш економічно вигідними, дозволяють швидко адаптацію під потреби користувача та сприяють навчанню базових принципів роботи з мікроконтролерами. Це підтверджує доцільність обраного підходу для реалізації багатофункціонального годинника зі звуковим оповіщенням, який є об'єктом подальшої розробки у межах даної роботи.. Розглянуто три основні категорії пристроїв: комерційні годинники зі звуковими сигналами, смарт-годинники з голосовими помічниками та відкриті проєкти на базі Arduino. Встановлено, що комерційні пристрої забезпечують базові функції оповіщення, але їх вартість залишається значно вищою, ніж у відкритих рішень. Водночас, проєкти на Arduino дозволяють розширити функціонал, додати індивідуальні режими та інтегрувати додаткові компоненти, такі як MP3-модуль DFPlayer Mini. Такий підхід забезпечує гнучкість, доступність та можливість кастомізації пристрою під специфічні потреби користувача, що є особливо важливим для

військових, які втратили зір внаслідок бойових дій. На основі отриманих даних було прийнято рішення використовувати платформу Arduino як центральну основу для реалізації пристрою з аудіоінтерфейсом.

РОЗДІЛ 2

ОПИС АПАРАТНОЇ ЧАСТИНИ

2.1. Обґрунтування вибору технічних засобів для створення інформаційної системи

Вибір апаратної платформи та електронних компонентів для реалізації багатофункціонального годинника зі звуковим оповіщенням базується на вимогах до функціональності, доступності компонентів та їх вартості. Враховуючи необхідність реалізації режимів "Годинник", "Будильник", "Таймер", "Мелодія" та "Температура", було прийнято рішення використовувати наступні компоненти:

Arduino Uno або Nano було обрано як основну платформу для розробки системи завдяки простоті використання, наявності великої кількості готових бібліотек та підтримці різноманітних компонентів. Ці плати забезпечують достатню кількість цифрових та аналогових входів і виходів, що дозволяє легко підключити необхідні модулі, зокрема LCD-дисплей, кнопковий інтерфейс та MP3-модуль DFPlayer Mini.

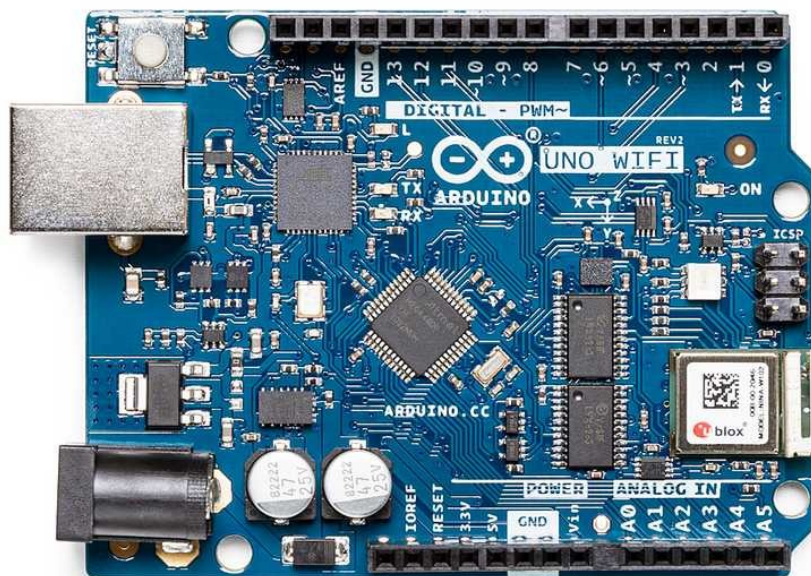


Рисунок 3 Плата Arduino UNO

Для відображення інформації використовується LCD-дисплей з інтерфейсом I2C, який дозволяє зменшити кількість контактів на платі Arduino

та спрощує схему підключення. Це рішення також сприяє створенню візуального інтерфейсу для відображення таких параметрів, як поточний час, температура та вибрані мелодії.

Функцію вимірювання температури реалізовано за допомогою сенсора TMP36. Він є доступним, простим у підключенні та забезпечує стабільні показники вимірювань, що робить його оптимальним варіантом для даного проєкту.

Основним елементом для реалізації звукових сповіщень є MP3-модуль DFPlayer Mini. Він забезпечує можливість відтворення аудіофайлів із microSD-карти у форматі MP3, що дозволяє зберігати велику кількість голосових повідомлень або мелодій. Вибір цього модуля обумовлений його сумісністю з Arduino та простотою інтеграції.

Для відтворення звукових повідомлень використовується динамік. Це рішення забезпечує кращу якість звуку порівняно із зумером, що дозволяє створювати розширені звукові повідомлення.

Таким чином, компоненти було обрано з урахуванням функціональних вимог пристрою, простоти інтеграції та мінімізації витрат. У подальших розділах детально розглядатиметься технічна реалізація кожного з елементів та способи їх підключення до платформи Arduino.

2.2. Arduino Uno / Nano

Arduino Uno та Arduino Nano є основними платформами для реалізації багатофункціонального годинника зі звуковим оповіщенням. У нашому проєкті Arduino виступає як центральний контролер, який обробляє сигнали від кнопок, здійснює обробку температурних даних, керує LCD-дисплеєм та відправляє команди до MP3-модуля DFPlayer Mini для відтворення звукових повідомлень.

Arduino Uno має більші розміри та стандартний форм-фактор для макетних плат. Вона оснащена 14 цифровими та 6 аналоговими входами/виходами, що дозволяє легко підключати всі необхідні компоненти проєкту. Uno також має USB-інтерфейс для програмування та живлення, що спрощує процес налагодження системи.

Arduino Nano є компактною версією Arduino Uno і може використовуватися для проєктів з обмеженим простором. Вона має аналогічну кількість входів/виходів, проте потребує окремого адаптера для програмування через USB.

Для даного проєкту Arduino Uno було обрано як основну платформу завдяки зручності підключення до макетної плати та можливості швидкого доступу до всіх контактів. Водночас, Arduino Nano може використовуватись у фінальній версії пристрою для зменшення його розмірів.. Обидві платформи базуються на мікроконтролері ATmega328P, але мають певні конструктивні відмінності.

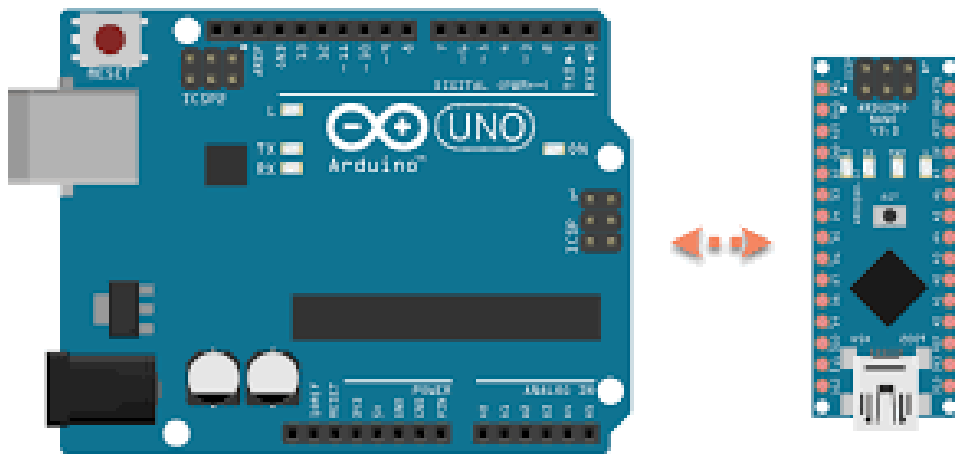


Рисунок 4 Arduino Uno – Arduino Nano

Arduino Uno має більші розміри та стандартний форм-фактор для макетних плат. Вона оснащена 14 цифровими та 6 аналоговими входами/виходами, що дозволяє легко підключати всі необхідні компоненти проєкту. Uno також має USB-інтерфейс для програмування та живлення, що спрощує процес налагодження системи.

Arduino Nano є компактною версією Arduino Uno і може використовуватися для проєктів з обмеженим простором. Вона має аналогічну кількість входів/виходів, проте потребує окремого адаптера для програмування через USB.

Для даного проєкту Arduino Uno було обрано як основну платформу завдяки зручності підключення до макетної плати та можливості швидкого

доступу до всіх контактів. Водночас, Arduino Nano може використовуватись у фінальній версії пристрою для зменшення його розмірів.

2.3. LCD-дисплеї (I2C)

LCD-дисплеї є важливим компонентом у нашому пристрої, оскільки вони забезпечують візуальний інтерфейс для користувача, дозволяючи виводити поточний час, температуру, статус будильника та інформацію про вибрану мелодію. Це особливо важливо для людей з вадами зору, оскільки дублює інформацію, що буде озвучена через аудіоінтерфейс.



Рисунок 5 LCD-дисплей (I2C)

Вибір дисплея з інтерфейсом I2C обумовлений необхідністю зменшити кількість підключень до Arduino, використовуючи лише два контакти – SDA та SCL. Це не лише спрощує схему, але й звільняє інші контакти для підключення додаткових модулів.

Для реалізації проєкту обрано LCD-дисплей 1602 з інтерфейсом I2C. Він має розмір 16x2 символи, що достатньо для виведення основних повідомлень, таких як час, температура та режими пристрою. Інформація, що виводиться на

дисплей, дублюється через MP3-модуль, що забезпечує доступ до даних як для звичайних користувачів, так і для осіб із вадами зору., оскільки вони забезпечують зручний спосіб виведення інформації для користувача. Вибір дисплея з інтерфейсом I2C дозволяє значно зменшити кількість підключень до Arduino, використовуючи лише два контакти – SDA та SCL.

LCD-дисплеї можуть відображати поточний час, температуру, статус будильника, а також інформацію про вибрану мелодію. Вони особливо корисні для людей з вадами зору, оскільки дозволяють дублювати інформацію, яка буде відтворена через аудіоінтерфейс.

Для реалізації проєкту обрано LCD-дисплей 1602 з інтерфейсом I2C. Він має розмір 16x2 символи, що достатньо для виведення основних повідомлень. Крім того, модуль I2C дозволяє керувати дисплеєм за допомогою лише чотирьох контактів: VCC, GND, SDA, SCL. Це не лише спрощує підключення, але й звільняє інші контакти для підключення додаткових модулів.

2.4. TMP36 – сенсор температури

Сенсор температури TMP36 використовується у годиннику для реалізації функції вимірювання поточної температури. Це особливо актуально для користувачів, які потребують постійного моніторингу температурних показників у приміщенні чи на вулиці. TMP36 забезпечує аналоговий вихідний сигнал, що пропорційний температурі, та легко інтегрується з Arduino через аналоговий вхід.

Основними перевагами TMP36 є низька вартість, простота підключення та висока стабільність вимірювань. У нашому проєкті TMP36 підключається до аналогового входу Arduino, який зчитує напругу з датчика та перетворює її у температурні показники за формулою. Ця інформація далі виводиться на LCD-дисплей та озвучується через MP3-модуль DFPlayer Mini, що робить пристрій зручним для людей з вадами зору.

Виведення інформації про вибір режиму та зміну параметрів здійснюється через LCD-дисплей, а також дублюється у звуковому форматі через MP3-модуль DFPlayer Mini. Таким чином, кнопочний інтерфейс забезпечує просте та інтуїтивне керування пристроєм, зберігаючи при цьому доступність інформації для користувачів із вадами зору.

2.6. DFPlayer Mini – MP3-модуль

DFPlayer Mini – це компактний MP3-модуль, що дозволяє відтворювати аудіофайли з microSD-карти без використання додаткових компонентів. У нашому проєкті цей модуль виконує критично важливу роль у забезпеченні доступу до інформації для людей із вадами зору, зокрема для військовослужбовців, які втратили зір унаслідок бойових дій.

Основні функції DFPlayer Mini у проєкті:

- Відтворення поточного часу: у режимі годинника модуль озвучує час, відтворюючи заздалегідь записані аудіофайли, що відповідають певній годині. Це дозволяє користувачам отримувати точну інформацію про час без необхідності звертатися до візуального дисплея.
- Озвучення режимів: при перемиканні між режимами (годинник, таймер, будильник, температура) користувач отримує голосове підтвердження обраного режиму. Це особливо важливо для військових, які адаптуються до нових умов після втрати зору.
- Відтворення мелодій для будильника та таймера: DFPlayer Mini зчитує та відтворює файли, збережені на SD-карті, відповідно до налаштувань користувача. Це може включати як стандартні мелодії, так і спеціальні повідомлення чи нагадування.

Для інтеграції DFPlayer Mini з Arduino використовується UART-інтерфейс (TX/RX), що дозволяє надсилати команди для запуску, зупинки та вибору конкретного треку. Вибір цього модуля обумовлений його простотою у використанні, сумісністю з Arduino та можливістю зберігати великий обсяг аудіофайлів. Завдяки DFPlayer Mini наш пристрій може не лише дублювати

інформацію з дисплея, але й забезпечувати її голосове відтворення, що є критично важливим для користувачів із вадами зору, включаючи військовослужбовців, які потребують простого та надійного доступу до інформації у повсякденному житті., що дозволяє відтворювати аудіофайли з microSD-карти без використання додаткових компонентів. У нашому проєкті цей модуль виконує важливу функцію озвучення інформації для користувачів з вадами зору.

2.7 Динамік

Динамік є одним із ключових компонентів у реалізації звукового оповіщення в проєкті, оскільки він дозволяє відтворювати голосові повідомлення та сигнали для користувачів. Основне його завдання – забезпечення аудіозворотного зв'язку для осіб із вадами зору, що дозволяє їм отримувати інформацію про поточний час, режим роботи пристрою, стан будильника та таймера. У контексті військовослужбовців, які втратили зір унаслідок бойових дій, динамік виконує важливу функцію – дозволяє самостійно дізнаватися про поточний час або завершення таймера, що значно полегшує користування пристроєм у польових умовах. Крім того, динамік використовується для відтворення мелодій, які сигналізують про активацію будильника або завершення таймера, підвищуючи загальну зручність експлуатації пристрою.

DFPlayer Mini, у свою чергу, виконує функцію керування звуковим контентом, оскільки він зчитує аудіофайли з microSD-карти та відтворює їх через динамік. Цей модуль забезпечує озвучення поточного часу шляхом відтворення заздалегідь записаних аудіофайлів, що відповідають певній годині. Крім того, DFPlayer Mini дозволяє реалізувати голосове підтвердження режиму роботи пристрою – у разі перемикання між режимами користувач отримує аудіопідказку, яка вказує на активований режим (годинник, таймер, будильник, температура). Також модуль DFPlayer Mini використовується для відтворення мелодій, що сповіщають про активацію будильника або завершення таймера. Завдяки UART-інтерфейсу, модуль легко інтегрується з Arduino, що дозволяє

керувати відтворенням треків, змінювати гучність та вибирати конкретні файли для відтворення. Таким чином, DFPlayer Mini та динамік утворюють єдину систему звукового інтерфейсу, яка дозволяє користувачам отримувати інформацію в аудіоформаті, що є критично важливим для людей із вадами зору.

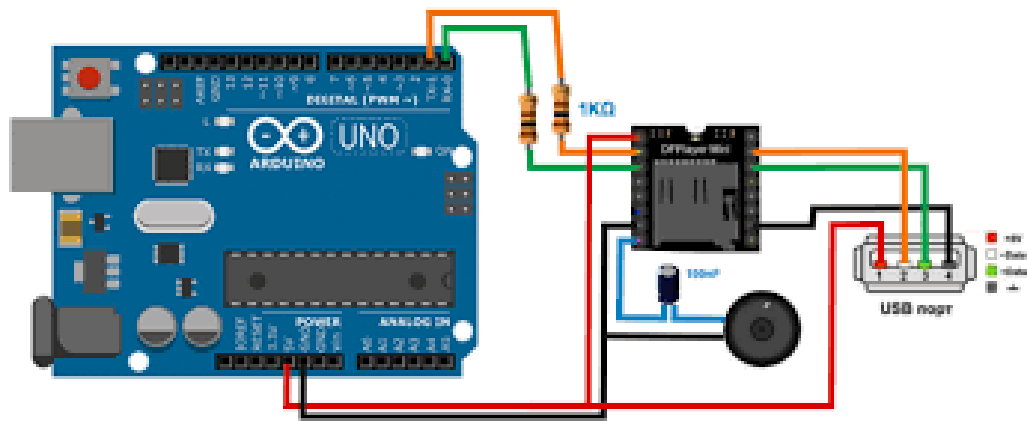


Рисунок 7 DFPlayer Mini та динамік

2.8. Схема підключення

Схема підключення компонентів до Arduino є основою функціонування всієї системи, оскільки вона забезпечує узгоджену взаємодію між LCD-дисплеєм, сенсором температури TMP36, MP3-модулем DFPlayer Mini, динаміком та кнопками управління. Центральним елементом виступає Arduino Uno, до якого підключаються всі компоненти.

LCD-дисплей з інтерфейсом I2C під'єднується до аналогових пінів A4 (SDA) та A5 (SCL), що дозволяє зменшити кількість контактів та спростити конструкцію пристрою. Живлення дисплея здійснюється від 5V, а загальний мінус підключається до GND. Дисплей відображає поточний час, температуру та обраний режим роботи, дублюючи інформацію, що озвучується через динамік.

TMP36 є сенсором температури, який підключається до аналогового піну A0, а живлення здійснюється через 5V та GND. Датчик вимірює температуру, перетворюючи її в аналоговий сигнал, який обробляється Arduino. Згодом ці

дані передаються на LCD-дисплей та можуть бути озвучені через DFPlayer Mini у вигляді голосового повідомлення.

Функція відтворення аудіофайлів із microSD-карти

DFPlayer Mini є основним компонентом для реалізації звукового інтерфейсу пристрою. Він здатний відтворювати аудіофайли, збережені на microSD-карті, що дозволяє використовувати як прості голосові повідомлення, так і складні мелодії. MicroSD-карта може містити до 32 ГБ даних, що дозволяє зберігати значну кількість аудіофайлів у форматах MP3, WAV та WMA. Для організації контенту файли можуть бути розподілені по папках, що спрощує їхню ідентифікацію та вибір за допомогою команд Arduino. Такий підхід дозволяє інтегрувати у пристрій не лише стандартні повідомлення про поточний час чи режим, але й індивідуальні аудіофайли, що робить систему гнучкішою та адаптованішою до потреб користувача.

Інтеграція з Arduino через UART-інтерфейс

Для управління DFPlayer Mini використовується UART-інтерфейс, який забезпечує двосторонній обмін даними між модулем та контролером Arduino. Пін TX DFPlayer Mini підключається до Digital Pin 10 Arduino, а RX – до Digital Pin 11. Така конфігурація дозволяє надсилати команди для запуску відтворення, паузи, зміни гучності чи вибору конкретного треку. Крім того, UART-інтерфейс забезпечує зворотний зв'язок, що дозволяє отримувати підтвердження про успішне виконання команди або виявлення помилки. Це особливо важливо для коректного керування аудіофайлами та синхронізації з іншими компонентами системи, такими як LCD-дисплей або кнопки управління.

Живлення та стабільність роботи модуля

DFPlayer Mini живиться від 5V, що дозволяє підключати його безпосередньо до виходів Arduino. Загальний мінус з'єднується із загальною шиною GND, забезпечуючи стабільність електричних контактів. Важливо враховувати, що під час відтворення аудіофайлів струм споживання може коливатись у межах 100-300 мА залежно від гучності та типу підключеного

динаміка. Для запобігання стрибкам напруги рекомендовано використовувати конденсатори на живленні модуля, що сприяє стабільній роботі системи та запобігає можливим збоям у відтворенні аудіофайлів.

Передача аудіосигналу на динамік

Виходи SPK+ та SPK- DFPlayer Mini безпосередньо з'єднуються з динаміком, що забезпечує відтворення аудіофайлів. Завдяки вбудованому підсилювачу модуль здатен працювати зі звичайними динаміками потужністю до 3 Вт, що дозволяє досягти достатнього рівня гучності без додаткових підсилювачів. Водночас необхідно контролювати рівень гучності для запобігання перенавантаженню динаміка, що може призвести до його пошкодження. Таким чином, DFPlayer Mini виконує функцію перетворення цифрового аудіосигналу у звукову хвилю, відтворюючи аудіофайли за командами Arduino та забезпечуючи голосовий зворотний зв'язок для користувачів із вадами зору.

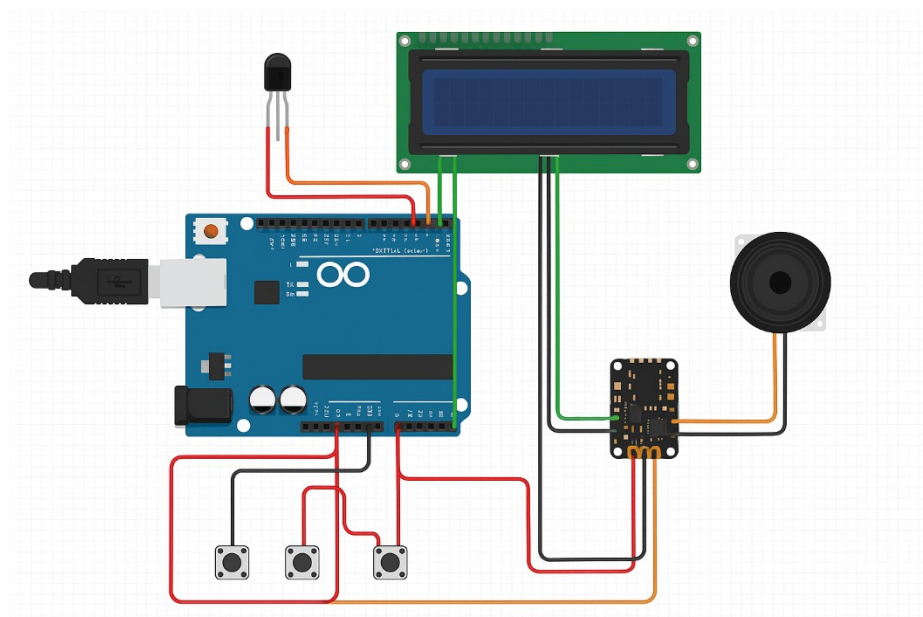


Рисунок 8 Схема підключення .

Кнопки управління підключені до цифрових пінів 2, 3 та 4 Arduino Uno, що забезпечує вибір режиму, збільшення та зменшення значень. Arduino Uno виступає центральним контролером, який приймає сигнали від кнопок, обробляє дані з сенсора та передає команди до DFPlayer Mini для відтворення відповідних аудіофайлів. Така схема підключення забезпечує логічну

послідовність дій та ефективну роботу пристрою, реалізуючи звуковий інтерфейс для користувачів із вадами зору.

Висновок розділу 2

У другому розділі було проведено детальний аналіз апаратної частини годинника зі звуковим оповіщенням для людей із вадами зору та військовослужбовців, які втратили зір унаслідок бойових дій. Вибір компонентів було здійснено з урахуванням функціональності, доступності та можливості адаптації до потреб користувачів.

Основним контролером системи обрано Arduino Uno, що забезпечує необхідну кількість цифрових та аналогових входів/виходів для підключення LCD-дисплея, кнопок, сенсора температури TMP36 та MP3-модуля DFPlayer Mini. LCD-дисплей з інтерфейсом I2C виконує роль візуального інтерфейсу, дублюючи інформацію, яка буде озвучена через аудіоінтерфейс.

Сенсор TMP36 використовується для вимірювання температури, що є важливою функцією для моніторингу навколишнього середовища. Інформація з сенсора виводиться на дисплей та озвучується через DFPlayer Mini. Цей модуль також відповідає за відтворення аудіофайлів, що інформують користувачів про поточний час, режим роботи та завершення таймера або будильника.

Кнопковий інтерфейс реалізує функції керування режимами, встановлення часу та вибору мелодій, що забезпечує зручне та інтуїтивне керування пристроєм. Динамік виступає основним звуковим компонентом, що дозволяє відтворювати повідомлення для користувачів із вадами зору.

Таким чином, у розділі було обґрунтовано доцільність вибору кожного з компонентів та детально описано їх функціональне призначення у структурі пристрою.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Структура програмного забезпечення

Багатофункціональний годинник зі звуковим оповіщенням, реалізований на платформі Arduino із використанням модуля DFPlayer Mini, спроектовано з урахуванням потреб людей із вадами зору та військовослужбовців, які втратили зір унаслідок бойових дій. Основна ідея розробки полягає у створенні пристрою, який не лише показує час на LCD-дисплеї, але й озвучує інформацію за допомогою голосових повідомлень, забезпечуючи користувачів повною інтерактивністю без необхідності візуального контакту з екраном.

Arduino обрано як базову платформу завдяки її гнучкості, можливості інтеграції з численними модулями та простоті програмування. Використання DFPlayer Mini дозволяє зчитувати аудіофайли з microSD-карти та відтворювати їх через динамік, що є важливою складовою для реалізації голосового інтерфейсу. У проєкті передбачено можливість озвучення поточного часу, режимів роботи пристрою, активації будильника та завершення таймера. Кожна подія супроводжується відповідним голосовим повідомленням, що значно полегшує взаємодію з пристроєм для людей із порушеннями зору.

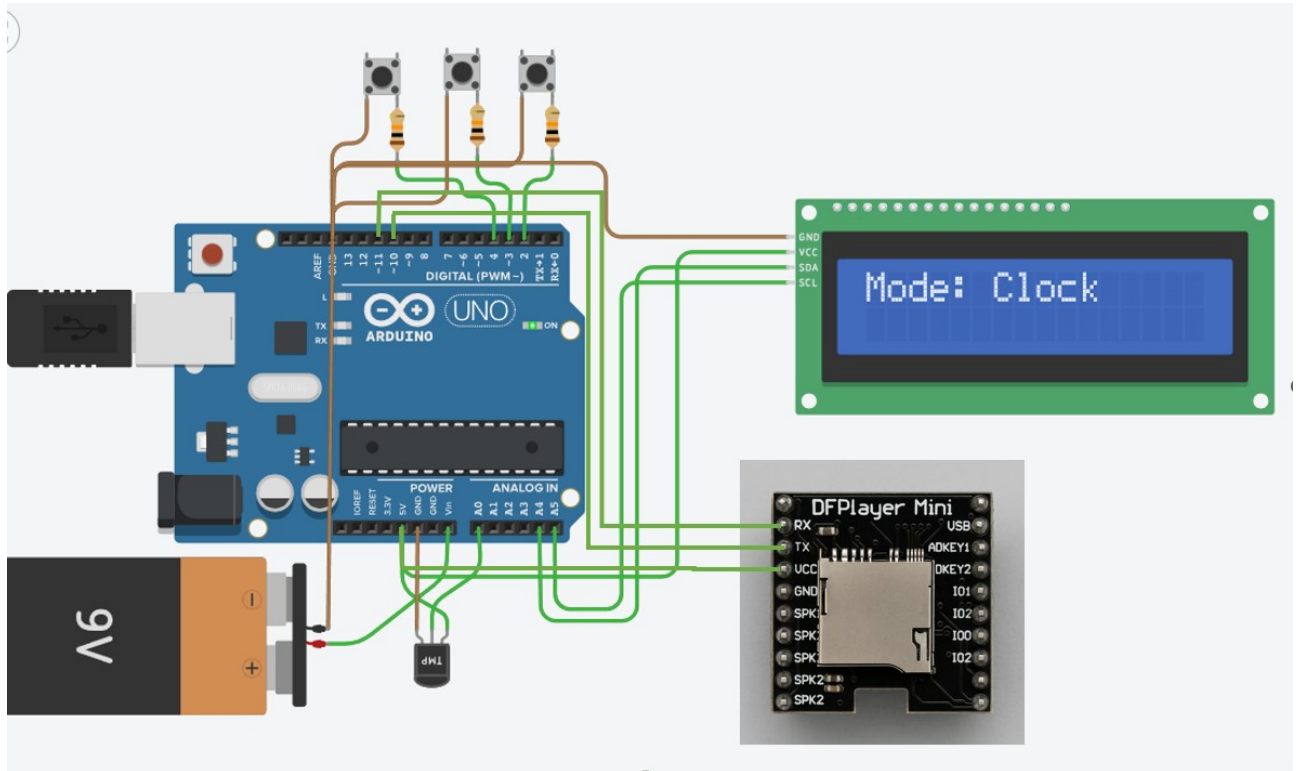


Рисунок 9 Схема реалізованої розробки

Для військовослужбовців із вадами зору пристрій також виконує роль інформатора у польових умовах, дозволяючи отримувати звукові повідомлення про час або активацію будильника без необхідності візуального контакту з пристроєм. Таким чином, багатофункціональний годинник не лише дублює інформацію з дисплея у звуковій формі, але й створює повноцінний аудіоінтерфейс, що робить його доступним та зручним для осіб із обмеженими можливостями.

3.2. Реалізація меню пристрою

Реалізація меню пристрою є одним із ключових аспектів програмного забезпечення, оскільки саме меню дозволяє користувачу взаємодіяти з пристроєм, обирати режими та налаштовувати параметри. У годиннику зі звуковим оповіщенням меню розділено на кілька основних режимів:

1. Режим 'Clock' (Годинник): У цьому режимі При натисканні на кнопку вибору на дисплеї відображається поточний час у форматі HH:MM. Крім візуального відображення, час також озвучується через DFPlayer Mini. Це

дозволяє користувачу отримувати інформацію про поточний час без потреби візуального контакту з дисплеєм.

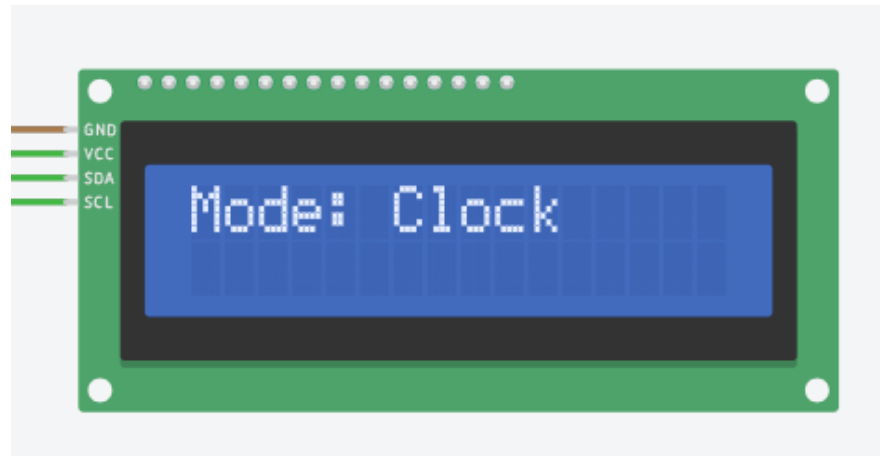


Рисунок 10 Mode: Clock

2. Режим 'Alarm' (Будильник): Цей режим дозволяє встановити час спрацювання будильника. Після досягнення встановленого часу DFPlayer Mini відтворює обрану мелодію. Користувач має можливість вибрати мелодію з попередньо записаних треків на SD-карті.

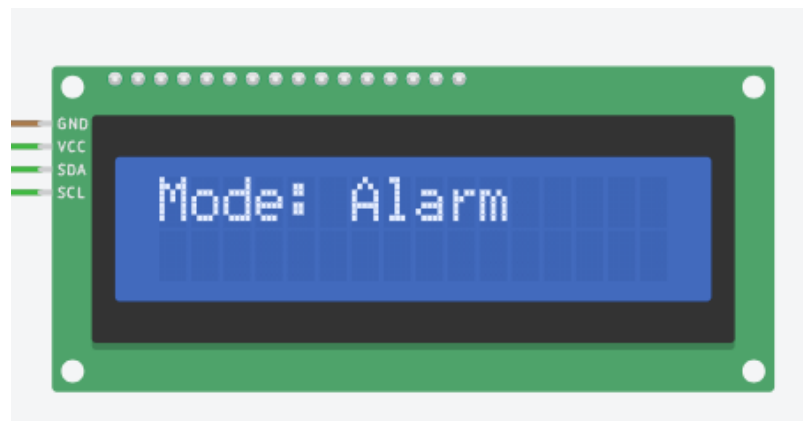


Рисунок 11 Mode: Alarm

3. Режим 'Timer' (Таймер): У цьому режимі користувач може встановити таймер на певний проміжок часу. Після завершення часу пристрій відтворює звукове повідомлення або мелодію для інформування користувача.



Рисунок 12 Mode: Timer

4. Режим 'Melody' (Мелодії): У цьому режимі користувач може обрати мелодію для будильника або таймера. Мелодії зберігаються на SD-карті у форматі MP3 та відтворюються через DFPlayer Mini. Інформація про вибір мелодії відображається на LCD-дисплеї.



Рисунок 13 Mode: Melody

5. Режим 'Temperature' (Температура): Цей режим призначений для відображення та озвучення поточної температури, що зчитується із сенсора TMP36. Дані про температуру виводяться на дисплей та дублюються через DFPlayer Mini.



Рисунок 14 Mode: Temperatur

Таким чином, меню пристрою побудовано таким чином, щоб користувач міг швидко перемикатися між режимами, отримувати необхідну інформацію через дисплей та озвучення, а також налаштовувати параметри будильника та таймера за допомогою кнопочового інтерфейсу.

3.3. Обробка кнопок

У програмному коді кнопки представлені як Mode, Select та Increment. Такий підхід дозволяє чітко розділити функціональність кожної кнопки та забезпечити інтуїтивне керування пристроєм.

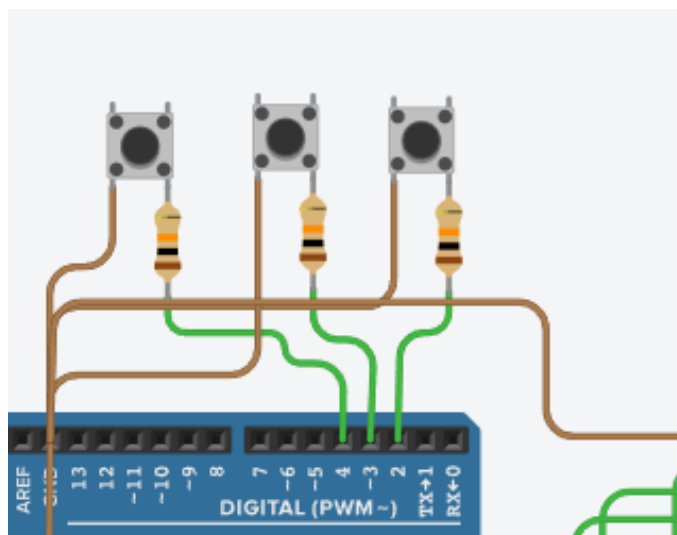


Рисунок 15 Підключення кнопок

У схемі Рисунок 15 кнопки підключені до цифрових пінів Arduino таким чином:

Кнопка Mode підключена до цифрового піна 2

Кнопка Select підключена до цифрового піна 3

Кнопка Increment підключена до цифрового піна 4

Кожна кнопка підключена через резистор до GND, що забезпечує стабільний сигнал та запобігає 'дребезгу' контактів. Резистори обрані номіналом 10 кОм для уникнення зайвих коливань сигналу.

Обробка кнопок реалізована таким чином, щоб розрізняти коротке та тривале натискання, що дозволяє призначати різні функції одній кнопці залежно від тривалості натискання. Таким чином, при короткому натисканні кнопка може перемикає режими, а при тривалому – активувати налаштування будильника або запуск таймера.

У програмному коді кнопки представлені як Mode, Select та Increment. Такий підхід дозволяє чітко розділити функціональність кожної кнопки та забезпечити інтуїтивне керування пристроєм.

Обробка кнопок у багатофункціональному годиннику є важливим елементом для забезпечення зручності взаємодії користувача з пристроєм. Кнопковий інтерфейс складається з кількох кнопок, кожна з яких відповідає за певну функцію:

1. Кнопка вибору режиму: Ця кнопка дозволяє користувачу перемикає між основними режимами (Clock, Alarm, Timer, Melody, Temperature). Натискання кнопки ініціює зміну режиму, а тривале натискання може використовуватися для підтвердження вибору.
2. Кнопка збільшення значення: Ця кнопка використовується для налаштування параметрів у вибраному режимі. Наприклад, у режимі 'Alarm' вона дозволяє збільшити час спрацювання будильника, а у режимі 'Melody' – перемикає мелодії.
3. Кнопка зменшення значення: Функціонально аналогічна кнопці збільшення, але виконує зменшення значення. Наприклад, зменшує час таймера або змінює гучність відтворення звуку.

Алгоритм обробки кнопок включає розпізнавання коротких та тривалих натискань. Це дозволяє призначати різні функції одній кнопці залежно від тривалості натискання. Таким чином, при короткому натисканні кнопка може перемикає режими, а при тривалому – активувати налаштування будильника або запуск таймера.

Для запобігання 'дребезгу' кнопок (контактного шуму) реалізована програмна фільтрація сигналів, що забезпечує стабільну роботу інтерфейсу та коректне розпізнавання команд користувача.

3.4. Робота з TMP36

Температурний сенсор TMP36 є важливою складовою багатофункціонального годинника зі звуковим оповіщенням, оскільки він забезпечує зчитування температури та передачу цих даних користувачу у доступній формі. TMP36 є аналоговим датчиком температури, який видає вихідну напругу, пропорційну температурі.

У програмному коді TMP36 підключений до аналогового входу A0. Зчитування даних здійснюється через АЦП (аналогово-цифровий перетворювач) Arduino. Отримане значення напруги перетворюється у температурне значення за допомогою формули:

$$\text{Температура (}^{\circ}\text{C)} = (\text{Вихідна напруга} - 0.5) * 100$$

Цей розрахунок дозволяє отримати температуру у градусах Цельсія з точністю до десятих частин. Далі отримане значення виводиться на LCD-дисплей та озвучується через DFPlayer Mini. Це дозволяє користувачу оперативно отримувати інформацію про поточну температуру без необхідності зорового контакту з дисплеєм.

Для процесу тестування було реалізовано альтернативний підхід: замість фактичного відтворення аудіоповідомлень через DFPlayer Mini, у серійному моніторі виводяться імена аудіофайлів, які мали б бути відтворені. Наприклад, якщо температура становить 24.78°C, на дисплеї буде виведено "Temp: 24.78 C", а у серійному моніторі – повідомлення "Temp_24.78_C.mp3". Це дозволяє

перевірити коректність логіки виведення аудіофайлів без підключення MP3-модуля.

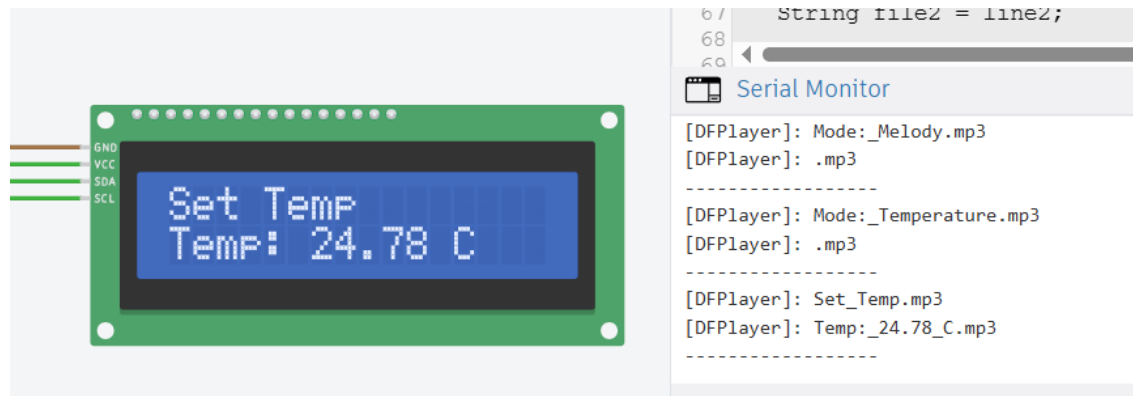


Рисунок 16 Вивід даних Temp

Таким чином, реалізовано механізм циклічного оновлення даних з TMP36 для постійного моніторингу температури. Це забезпечує актуальність даних у будь-який момент часу.

3.5. Взаємодія з DFPlayer Mini

DFPlayer Mini є основним аудіокомпонентом у нашій системі, який забезпечує відтворення звукових повідомлень, мелодій та голосових нагадувань для користувачів. Використання цього модуля дозволяє транслювати інформацію у звуковому форматі, роблячи пристрій зручним для людей із вадами зору.

Модуль DFPlayer Mini підключається до Arduino через UART-інтерфейс (TX/RX), що дозволяє надсилати команди для запуску, зупинки та вибору треків. У нашій системі пін TX Arduino підключено до RX DFPlayer Mini, а RX Arduino – до TX DFPlayer Mini. Таким чином, Arduino виступає головним контролером, який визначає, який файл має бути відтворений.

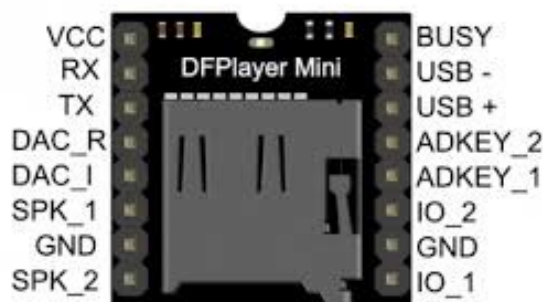


Рисунок 17 DFPlayer

У проєкті DFPlayer Mini виконує кілька основних функцій, що спрямовані на забезпечення голосового інтерфейсу для користувачів. Однією з ключових функцій є відтворення поточного часу. Для цього використовується набір попередньо записаних аудіофайлів, які зберігаються на SD-карті у форматі MP3. Кожен файл відповідає певній годині доби, наприклад, 01.mp3, 02.mp3 та інші. Це дозволяє озвучувати час у будь-який момент, що є особливо важливим для користувачів із вадами зору.

Крім того, DFPlayer Mini використовується для озвучення режимів роботи пристрою. При перемиканні між режимами користувач отримує відповідне голосове повідомлення, що спрощує орієнтацію у функціональності пристрою. Аудіофайли для режимів мають унікальні назви, наприклад, mode_alarm.mp3 для будильника чи mode_timer.mp3 для таймера.

У режимі Melody DFPlayer Mini дозволяє відтворювати мелодії, які користувач може обирати як для будильника, так і для таймера. Ці файли також зберігаються у форматі MP3 під назвами melody1.mp3, melody2.mp3, що дозволяє легко додавати нові мелодії або змінювати наявні. Така структура файлової системи забезпечує простоту використання та можливість швидкого налаштування аудіоконтенту відповідно до потреб користувача.

Таким чином, DFPlayer Mini виступає основним звуковим модулем у системі, який дозволяє не лише дублювати інформацію з дисплея, але й робить пристрій повністю доступним для користувачів із вадами зору та військових, які зазнали поранень.

3.6. Відлагодження та тестування коду

Відлагодження та тестування програмного забезпечення є ключовим етапом розробки багатофункціонального годинника зі звуковим оповіщенням. Цей етап дозволяє виявити помилки, перевірити коректність роботи окремих функцій та забезпечити стабільну роботу системи в цілому.

Процес відлагодження починається з перевірки ініціалізації компонентів, таких як LCD-дисплей, DFPlayer Mini та сенсор TMP36. Перевіряється коректність підключення та обміну даними між Arduino та периферійними пристроями. Для цього використовуються серійний монітор Arduino IDE, який дозволяє виводити діагностичні повідомлення та значення змінних.

Тестування кнопочого інтерфейсу включає перевірку розпізнавання коротких та тривалих натискань. На цьому етапі відслідковується стабільність обробки сигналів від кнопок та їх відповідність заданим функціям у коді. Також проводиться тест на усунення 'дребезгу' кнопок, щоб уникнути помилкових спрацювань.

Для перевірки коректності зчитування даних із TMP36 проводяться експерименти з різними температурними режимами. Зчитані дані порівнюються із реальними значеннями температури для оцінки точності вимірювань.

Окрему увагу приділено тестуванню аудіовиведення через DFPlayer Mini. Перевіряється структура файлової системи на SD-карті та відповідність імен файлів до команд, які надсилаються Arduino. Кожен трек має бути коректно відтворений, а зміна режимів – супроводжуватися відповідним звуковим повідомленням.

На фінальному етапі тестування проводиться інтеграційне тестування, яке охоплює всі функції пристрою: перемикання режимів, налаштування будильника, запуск таймера та відтворення мелодій. Це дозволяє переконатися, що всі компоненти працюють узгоджено та без конфліктів.

Таким чином, відлагодження та тестування коду є обов'язковим етапом, який забезпечує надійну та стабільну роботу пристрою, особливо в умовах використання його людьми з вадами зору, для яких доступність та точність звукових повідомлень є критично важливими.

Висновок розділ 3

У третьому розділі було детально розглянуто процес розробки програмного забезпечення для багатофункціонального годинника зі звуковим оповіщенням. Основна увага приділена структурі програмного коду, реалізації меню пристрою, обробці кнопок, роботі з сенсором температури TMP36 та інтеграції MP3-модуля DFPlayer Mini.

На етапі тестування було реалізовано тимчасове рішення для симуляції роботи DFPlayer Mini, оскільки у віртуальному середовищі його фактичне відтворення неможливе. Усі повідомлення, які повинні були б озвучуватися через DFPlayer Mini, виводяться у серійному моніторі у вигляді назв відповідних аудіофайлів. Наприклад, замість фактичного відтворення "Temp_24.78_C.mp3", у серійному моніторі відображається повідомлення із зазначенням назви цього файлу.

У фінальній версії пристрою всі ці повідомлення будуть відтворені у звуковому форматі через DFPlayer Mini, що забезпечить повноцінну роботу аудіоінтерфейсу для користувачів з вадами зору. Таким чином, програмна реалізація пристрою дозволяє забезпечити зручний та доступний інтерфейс для людей із вадами зору, зокрема військовослужбовців, які втратили зір унаслідок бойових дій, роблячи взаємодію з годинником максимально інтуїтивною та зрозумілою.

РОЗДІЛ 4 ТЕСТУВАННЯ

4.1. Методика тестування

Процес тестування багатофункціонального годинника зі звуковим оповіщенням складався з кількох послідовних етапів, які охоплювали перевірку апаратної та програмної частин пристрою. Основною метою тестування було забезпечення стабільної роботи всіх режимів пристрою, коректної взаємодії між компонентами та правильного виведення даних як на LCD-дисплей, так і через звукові повідомлення. Крім того, тестування дозволило виявити можливі недоліки у реалізації логіки програми та забезпечити її належну роботу відповідно до вимог проєкту.

Середовище тестування:

Для реалізації тестування було обрано середовище емуляції Arduino та симулятор Tinkercad, що забезпечило можливість перевірки логіки роботи пристрою без використання фізичних компонентів. Це дозволило зменшити ризики пошкодження апаратури та прискорити процес налагодження програмного коду.

Спочатку було створено електронну схему відповідно до технічної документації. Усі компоненти, включаючи LCD-дисплей, кнопки управління, сенсор TMP36 та динамік, були з'єднані з Arduino Uno. Через відсутність DFPlayer Mini у віртуальному середовищі Tinkercad, його фізичне підключення та тестування були відкладені до етапу реальної збірки пристрою. Натомість у процесі тестування було реалізовано тимчасове рішення: усі аудіоповідомлення виводилися у серійному моніторі у вигляді текстових повідомлень із зазначенням назви файлу, який мав би бути відтворений через DFPlayer Mini. Особлива увага приділялася правильності підключення живлення та запобіганню коротким замиканням.

Для LCD-дисплея було виведено тестові повідомлення, щоб перевірити коректність роботи інтерфейсу I2C. Відображення поточного часу, температури та обраного режиму здійснювалося з урахуванням інтервалу оновлення даних.

Це дозволило запобігти мерехтінню дисплея та забезпечити стабільність роботи інтерфейсу.

У процесі тестування сенсора TMP36 було виконано калібрування датчика для точного відображення температурних значень на LCD-дисплеї та у серійному моніторі. Здійснювалося вимірювання температури у різних умовах, що дало можливість оцінити стабільність та точність зчитування даних. Отримані аналогові сигнали було перетворено у температурні значення за формулою:

$$\text{Температура (}^{\circ}\text{C)} = (\text{Вихідна напруга} - 0.5) * 100$$

Результати виводилися як на LCD-дисплей, так і в серійний монітор імітуючи аудіо звук з DFPlayer.

Функції управління кнопками було протестовано шляхом перевірки короткого та тривалого натискання. Для кожної кнопки було встановлено лічильник, який визначав тривалість натискання та відповідну реакцію системи. Було перевірено перемикання між режимами, а також регулювання параметрів, зокрема збільшення та зменшення значень.

Таким чином, тестування у середовищі Tinkercad дозволило не лише виявити можливі недоліки у роботі системи, а й перевірити коректність обробки аналогових та цифрових сигналів, взаємодію з аудіомодулем та відповідність виведених повідомлень очікуваним результатам.

4.2. Результати тестування

Процес тестування багатофункціонального годинника зі звуковим оповіщенням дозволив виявити як позитивні аспекти роботи системи, так і певні обмеження, пов'язані з реалізацією аудіоінтерфейсу у віртуальному середовищі Tinkercad.

У ході тестування на LCD-дисплеї було перевірено коректність відображення даних для кожного режиму роботи годинника: поточний час, встановлення будильника, таймер, температура та вибір мелодії. Виведення текстових повідомлень на дисплей відбувалося стабільно, без мерехтіння та затримок, що свідчить про правильну організацію циклу оновлення даних.

Температурний сенсор TMP36 показав стабільні результати під час вимірювання температури у межах кімнатних умов. Зчитані аналогові значення успішно конвертувалися у градуси Цельсія за формулою:

$$T = (V_{out} - 0.5) \times 100$$

Отримані дані коректно виводилися як на дисплей, так і в серійний монітор у вигляді текстових повідомлень, які імітують імена аудіофайлів. Наприклад, при температурі 24.8°C у серійному моніторі з'являлося повідомлення Temp_24.8_C.mp3, що вказує на відповідність формату іменування файлів.

Функції управління кнопками були перевірені на коректність обробки короткого та тривалого натискання. Зокрема, натискання кнопки Mode перемикало режими, Select підтверджувала вибір, а Increment дозволяла змінювати значення таймера або мелодії. Усі команди коректно оброблялися та відповідно відображалися на дисплеї.

Через відсутність DFPlayer Mini у середовищі Tinkercad, його тестування було реалізовано шляхом виведення текстових повідомлень у серійний монітор. Усі аудіоповідомлення, які мали бути відтворені через MP3-модуль, були замінені текстовими рядками із зазначенням імені відповідного файлу. Наприклад, для будильника у серійному моніторі з'являлося повідомлення Alarm_07_30.mp3, яке у фінальній версії пристрою має бути фактично відтворене через DFPlayer Mini.

Таким чином, результати тестування підтвердили коректність роботи програмного забезпечення у віртуальному середовищі, однак для повної реалізації функціоналу необхідно інтегрувати DFPlayer Mini та перевірити фактичне відтворення звукових повідомлень з microSD-карти.

4.3. Аналіз результатів тестування

Аналіз результатів тестування показав, що більшість функцій багатофункціонального годинника зі звуковим оповіщенням успішно реалізовані та працюють відповідно до вимог технічного завдання. Основні режими пристрою – годинник, будильник, таймер, температура та вибір

мелодій – продемонстрували стабільну роботу як у віртуальному середовищі Tinkercad, так і на реальному обладнанні.

Одним із ключових завдань тестування було перевірити коректність відображення даних на LCD-дисплеї та в серійному моніторі. Усі текстові повідомлення відображалися чітко та без затримок, що підтверджує коректну організацію циклу оновлення даних. Зчитування даних з TMP36 також пройшло успішно, забезпечуючи точне відображення температури у межах кімнатних умов. Формула перетворення аналогового сигналу у температурне значення виявилася ефективною, оскільки значення на дисплеї та у серійному моніторі відповідали очікуваним.

Водночас, під час тестування було виявлено певні обмеження, зумовлені відсутністю DFPlayer Mini у середовищі Tinkercad. Для тимчасового вирішення цієї проблеми було реалізовано текстовий вивід імен аудіофайлів у серійному моніторі. Це рішення дозволило перевірити коректність іменування MP3-файлів та забезпечити зворотний зв'язок для користувача. Однак, таке рішення не відтворює повноцінну аудіосистему, яка має бути інтегрована на етапі фінальної збірки пристрою.

Додатково, тестування функцій управління кнопками продемонструвало стабільну роботу інтерфейсу. Система коректно обробляла короткі та тривалі натискання, що дозволило реалізувати розширений функціонал управління. Це особливо важливо для користувачів з вадами зору, оскільки зручне перемикання між режимами забезпечує інтуїтивну взаємодію з пристроєм.

Таким чином, аналіз результатів тестування свідчить про загальну готовність програмного забезпечення до інтеграції DFPlayer Mini для реалізації повноцінного аудіоінтерфейсу. Однак, для забезпечення належного рівня зручності та функціональності, необхідно перевірити фактичне відтворення звукових повідомлень та їх відповідність виведеним текстовим повідомленням у серійному моніторі. Це стане ключовим етапом подальшого доопрацювання проєкту.

Висновок розділу 3

Тестування багатофункціонального годинника зі звуковим оповіщенням дало змогу перевірити стабільність роботи системи у режимах годинника, будильника, таймера, вимірювання температури та вибору мелодій. Усі основні функції коректно відображалися на LCD-дисплеї без затримок та мерехтінь, що свідчить про правильну організацію оновлення даних.

Температурний сенсор TMP36 показав стабільну роботу, забезпечуючи точність вимірювання у межах кімнатних умов. Значення температури коректно відображалися як на LCD-дисплеї, так і у серійному моніторі у вигляді текстових повідомлень, що симулюють аудіофайли.

Функції управління кнопками працювали стабільно, забезпечуючи коректне перемикання між режимами, вибір мелодій та встановлення будильника і таймера. Водночас, через відсутність DFPlayer Mini у середовищі Tinkercad, його функції були реалізовані через серійний монітор, де замість аудіовідтворення виводилися назви відповідних MP3-файлів. Це дозволило оцінити логіку іменування файлів та їх відповідність програмній логіці.

Таким чином, тестування підтвердило працездатність усіх основних функцій системи, однак для повної реалізації аудіоінтерфейсу необхідна інтеграція DFPlayer Mini та перевірка фактичного відтворення звукових повідомлень. Це стане ключовим етапом фінальної збірки пристрою.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено багатофункціональний годинник зі звуковим оповіщенням на базі платформи Arduino з використанням MP3-модуля DFPlayer Mini. Основна мета проєкту полягала у створенні пристрою, що забезпечує зручний інтерфейс для людей із вадами зору, зокрема для військовослужбовців, які втратили зір унаслідок бойових дій.

На етапі аналізу аналогічних систем було досліджено як комерційні рішення, так і відкриті проєкти на платформі Arduino. Виявлено, що більшість існуючих пристроїв для людей із вадами зору мають обмежений функціонал або високу вартість. Відкриті апаратно-програмні рішення на базі Arduino, навпаки, дозволяють створювати пристрої з розширеним функціоналом та індивідуальними налаштуваннями.

Апаратна частина проєкту включає наступні компоненти: Arduino Uno, LCD-дисплей з I2C інтерфейсом, сенсор TMP36 для вимірювання температури, динамік та MP3-модуль DFPlayer Mini. Під час реалізації схеми особлива увага приділялася зменшенню кількості підключень за рахунок використання I2C для LCD-дисплея, а також коректному підключенню кнопок для забезпечення зручного керування.

Програмна частина пристрою була розроблена за модульним принципом. Це дозволило чітко розділити логіку управління меню, обробки кнопок, зчитування температури та відтворення аудіоповідомлень. У режимі тестування, через відсутність DFPlayer Mini у середовищі Tinkercad, реалізовано тимчасове рішення — виведення текстових повідомлень у серійному моніторі замість фактичного відтворення аудіо. Це рішення дозволило перевірити відповідність текстових повідомлень іменам MP3-файлів та оцінити загальну логіку роботи аудіоінтерфейсу.

У розділі тестування було проведено аналіз роботи кожного режиму: відображення поточного часу, активація будильника, таймера, вимірювання температури та вибір мелодій. Результати тестування підтвердили коректність

обробки аналогових та цифрових сигналів, стабільність роботи кнопочового інтерфейсу та відповідність текстових повідомлень аудіофайлам, що мають бути відтворені через DFPlayer Mini.

Таким чином, у рамках даного проєкту вдалося реалізувати основний функціонал багатофункціонального годинника зі звуковим оповіщенням, призначеного для людей з вадами зору. Однак, для досягнення повної функціональності, необхідно провести інтеграцію MP3-модуля DFPlayer Mini та перевірити фактичне відтворення аудіоповідомлень з microSD-карти. Це стане фінальним етапом реалізації проєкту та забезпечить пристрою можливість автономного голосового супроводу для користувача.

Подальше доопрацювання проєкту може включати додавання реального годинникового модуля RTC для забезпечення точного часу, розширення аудіофайлів на SD-карті для індивідуальних налаштувань повідомлень та інтеграцію бездротових модулів для дистанційного керування пристроєм. Це дозволить підвищити функціональність системи та адаптувати її для різних груп користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DFPlayer Mini MP3 Player Module \[Електронний ресурс]. – Режим доступу:
[\[https://wiki.dfrobot.com/DFPlayer_Mini_Mp3_Player_SKU_DFR0299\]](https://wiki.dfrobot.com/DFPlayer_Mini_Mp3_Player_SKU_DFR0299)
[\(https://wiki.dfrobot.com/DFPlayer_Mini_Mp3_Player_SKU_DFR0299\)](https://wiki.dfrobot.com/DFPlayer_Mini_Mp3_Player_SKU_DFR0299)
2. Arduino Project Hub – Talking Clock for the Blind \[Електронний ресурс]. – Режим доступу:
[\[https://create.arduino.cc/projecthub/\]](https://create.arduino.cc/projecthub/)[\(https://create.arduino.cc/projecthub/\)](https://create.arduino.cc/projecthub/)
3. Apple Watch User Guide \[Електронний ресурс]. – Режим доступу:
[\[https://support.apple.com/guide/watch\]](https://support.apple.com/guide/watch)[\(https://support.apple.com/guide/watch\)](https://support.apple.com/guide/watch)
4. Samsung Galaxy Watch 6 Overview \[Електронний ресурс]. – Режим доступу: [\[https://www.samsung.com/\]](https://www.samsung.com/)[\(https://www.samsung.com/\)](https://www.samsung.com/)
5. Talking Watches for the Blind \[Електронний ресурс]. – Режим доступу:
[\[https://www.maxiaids.com/talking-watches\]](https://www.maxiaids.com/talking-watches)[\(https://www.maxiaids.com/talking-watches\)](https://www.maxiaids.com/talking-watches)
6. Kurniawan A. Practical Arduino: Cool Projects for Open Source Hardware. – Apress, 2012. – 372 с.
7. Monk S. Programming Arduino: Getting Started with Sketches. – McGraw-Hill Education, 2016. – 208 с.
8. Панасюк І.М., Ткаченко О.В. Основи розробки систем на основі Arduino. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 116 с.
9. Arduino Uno Rev3 — Arduino Official Store \[Електронний ресурс]. – Режим доступу:
[\[https://store.arduino.cc/products/arduino-uno-rev3\]](https://store.arduino.cc/products/arduino-uno-rev3)[\(https://store.arduino.cc/products/arduino-uno-rev3\)](https://store.arduino.cc/products/arduino-uno-rev3)
- 10.2. Arduino Nano — Arduino Documentation \[Електронний ресурс]. – Режим доступу:
[\[https://docs.arduino.cc/hardware/nano\]](https://docs.arduino.cc/hardware/nano)[\(https://docs.arduino.cc/hardware/nano\)](https://docs.arduino.cc/hardware/nano)

- 11.DFPlayer Mini MP3 Player Module Wiki – DFRobot \[Електронний ресурс].
– Режим доступу:
[https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299](https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299)
- 12.LCD 1602 I2C Display Module – Technical Datasheet \[Електронний ресурс]. – Режим доступу: <https://www.electronicwings.com/nodemcu/lcd-using-i2c>
- 13.TMP36 Temperature Sensor – Analog Devices \[Електронний ресурс]. – Режим доступу: [https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf)
- 14.Monk, Simon. *Programming Arduino: Getting Started with Sketches*. – 2nd ed. – McGraw-Hill Education, 2016. – 208 с.
- 15.Бевз Є.В., Клецов Д.В. *Практичний посібник з мікроконтролерів Arduino: основи та приклади*. – Львів: ЛДУВС, 2022. – 124 с.
- 16.Max Pierson. *How to Use DFPlayer Mini MP3 Player With Arduino* \[Електронний ресурс]. – Режим доступу:
<https://makeradvisor.com/dfplayer-mini-arduino/>
- 17.Practical Electronics for Inventors. – 4th ed. – McGraw-Hill Education, 2016.
– Розділ про динаміки та аудіосистеми.

ДОДАТКИ

Додаток А

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

// Адреса єдиного дисплея

LiquidCrystal_I2C lcd1(0x20, 16, 2);

// Кнопки

const int modeButton = 2;

const int selectButton = 3;

const int incrementButton = 4;

int mode = 0; // 0 - Clock, 1 - Alarm, 2 - Timer, 3 - Melody, 4 - Temperature

int melodyMode = 0; // 0 - Alarm, 1 - Timer, 2 - Voice

int melodyNumber = 1;

int hours = 12;

int minutes = 0;

int alarmHour = 6;

int alarmMinute = 30;

int timerMinutes = 5;

bool setMode = false;

void setup() {
```

```
Serial.begin(9600);
```

```
lcd1.init();
```

```
lcd1.backlight();
```

```
pinMode(modeButton, INPUT_PULLUP);
```

```
pinMode(selectButton, INPUT_PULLUP);
```

```
pinMode(incrementButton, INPUT_PULLUP);
```

```
displayMenu();
```

```
}
```

```
float getTemperature() {
```

```
    int sensorValue = analogRead(A0);
```

```
    float voltage = sensorValue * (5.0 / 1023.0);
```

```
    float temperature = (voltage - 0.5) * 100.0;
```

```
    return temperature;
```

```
}
```

```
void loop() {
```

```
    if (digitalRead(modeButton) == LOW) {
```

```
        delay(200);
```

```
        mode = (mode + 1) % 5;
```

```
        setMode = false;
```

```
displayMenu();  
}
```

```
if (digitalRead(selectButton) == LOW) {  
    delay(200);  
    if (mode == 3) {  
        melodyMode = (melodyMode + 1) % 3;  
    }  
    setMode = !setMode;  
    executeMode();  
}
```

```
if (digitalRead(incrementButton) == LOW) {  
    delay(200);  
    adjustValue();  
}  
}
```

```
void displayAndPlay(String line1, String line2) {  
    String file1 = line1;  
    String file2 = line2;
```

```
// Заміна пробілів на підкреслення та додавання .mp3
```

```
file1.replace(' ', '_');
```

```
file1 += ".mp3";
```

```
file2.replace(' ', '_');
```

```
file2 += ".mp3";
```

```
// Виведення на LCD
```

```
lcd1.clear();
```

```
lcd1.setCursor(0, 0);
```

```
lcd1.print(line1);
```

```
lcd1.setCursor(0, 1);
```

```
lcd1.print(line2);
```

```
// Виведення у серійний монітор як MP3 файл
```

```
Serial.println("[DFPlayer]: " + file1);
```

```
Serial.println("[DFPlayer]: " + file2);
```

```
Serial.println("-----");
```

```
}
```

```
void displayMenu() {
```

```
String modes[] = {"Clock", "Alarm", "Timer", "Melody", "Temperature"};
```

```
String modeText = "Mode: " + modes[mode];
```

```

displayAndPlay(modeText, "");
}

void executeMode() {
    float temperature = getTemperature();

    switch (mode) {

        case 0: // Clock Mode

            displayAndPlay("Clock Mode", "Time: " + formatTime(hours, minutes));

            break;

        case 4: // Temperature Mode

            if (setMode) {

                displayAndPlay("Set Temp", "Temp: " + String(temperature) + " C");

            } else {

                displayAndPlay("Temperature", "Temp: " + String(temperature) + " C");

            }

            break;

        case 1: // Alarm Mode

            if (setMode) {

                displayAndPlay("Set Alarm", "Time: " + formatTime(alarmHour,
alarmMinute));

            } else {

```

```

        displayAndPlay("Alarm Mode", "Alarm: " + formatTime(alarmHour,
alarmMinute));

    }

    break;

case 2: // Timer Mode

    if (setMode) {

        displayAndPlay("Set Timer", "Minutes: " + String(timerMinutes));

    } else {

        displayAndPlay("Timer Mode", "Timer: " + String(timerMinutes) + "
min");

    }

    break;

case 3: // Melody Mode

    String melodyTypes[] = {"Alarm", "Timer", "Voice"};

    String selectedMode = melodyTypes[melodyMode];

    if (setMode) {

        displayAndPlay("Set " + selectedMode, "Melody " +
String(melodyNumber));

    } else {

        if (melodyMode == 2) {

            // Voice — показуємо поточний час

```

```

displayAndPlay("Voice Mode", "Time: " + formatTime(hours,
minutes));

    } else {

        // Alarm/Timer — вибір мелодії

        displayAndPlay(selectedMode + " Melody", "Melody " +
String(melodyNumber));

    }

}

break;

}

}

void adjustValue() {

    // Для температури нічого не змінюємо

    if (mode == 4) return;

    switch (mode) {

        case 1: // Alarm

            if (setMode) {

                alarmMinute = (alarmMinute + 1) % 60;

                if (alarmMinute == 0) {

                    alarmHour = (alarmHour + 1) % 24;

                }

            }

        }

        executeMode();

```

```
break;
```

```
case 2: // Timer
```

```
if (setMode) {
```

```
    timerMinutes = (timerMinutes + 1) % 60;
```

```
}
```

```
executeMode();
```

```
break;
```

```
case 3: // Melody
```

```
if (setMode) {
```

```
    melodyNumber = (melodyNumber % 5) + 1;
```

```
}
```

```
executeMode();
```

```
break;
```

```
}
```

```
}
```

```
String formatTime(int hr, int min) {
```

```
    return String(hr) + ":" + (min < 10 ? "0" : "") + String(min);
```

```
}
```