

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютеризована система інформування про прийом медикаментів*

Виконав(ла): студент(ка) *IV* курсу, групи *СІ-41*

спеціальності *123 «Комп'ютерна інженерія»*

(шифр і назва спеціальності)

(підпис)

Бражніков В.А.

(прізвище та ініціали)

Керівник

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тим С.В.

(прізвище та ініціали)

Завідувач
кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Бражнікову Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система інформування про прийом медикаментів

Керівник роботи Осухівська Галина Михайлівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 28 » 02 2023 року № 4/7-238

2. Термін подання студентом завершеної роботи 22.06.2023

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз вимог до комп'ютеризованої системи інформування про прийом медикаментів, аналіз можливих рішень поставленого завдання, розробка узагальненої структури системи, обґрунтування вибору апаратного забезпечення, обґрунтування вибору програмного забезпечення, проектування комп'ютеризованої системи інформування про прийом медикаментів, реалізація проєктних рішень, тестування комп'ютеризованої системи інформування про прийом медикаментів, безпека життєдіяльності та основи охорони праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Структурна схема

Електрична принципова схема роботи комп'ютеризованої системи інформування про прийом

медикаментів

Блок-схема роботи комп'ютеризованої системи інформування про прийом медикаментів

Принципова схема проєкту

АНОТАЦІЯ

Комп'ютеризована система інформування про прийом медикаментів на основі Arduino UNO // Кваліфікаційна робота бакалавра // Бражніков Владислав Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ-41 // Тернопіль, 2023 // с. – 73, рис. – 25, табл. – 2, кресл. – 4, додат. – 3, бібліогр. – 15.

Ключові слова: КОМП'ЮТЕРИЗОВАНА СИСТЕМА, ARDUINO UNO, МІКРОКОНТРОЛЕР, НАГАДУВАННЯ ПРО ПРИЙОМ МЕДИКАМЕНТІВ,

Кваліфікаційна робота розроблена для створення системи, що дозволяє нагадувати користувачу про прийом ліків у заданий інтервал часу.

Кваліфікаційна робота складається з чотирьох розділів.

У першому розділі проведено аналіз технічного завдання, вимог до системи, та можливих рішень.

В другому розділі описано процес проєктування, аналіз вибору програмного та апаратного забезпечення.

В третьому розділі описана практична частина виконання проєкту – створення блок-схем алгоритмів, програмного коду та макету.

Четвертий розділ описує безпеку життєдіяльності та основи охорони праці.

ANNOTATION

Computerized system of informing about medication intake based on Arduino UNO // Bachelor's thesis // Brazhnikov Vladyslav Andriyivych // Ternopil National Technical University named after Ivan Puluj, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Networks, SI-41 group // Ternopil, 2023 // p. – 73, fig. – 25, table. – 2, chair. – 4, appendix. – 3, bibliogr. – 15.

Keywords: COMPUTERIZED SYSTEM, ARDUINO UNO, MICROCONTROLLER, MEDICATION REMINDER,

The qualification work is designed to create a system that allows you to remind the user to take medication at a given time interval.

The qualification work consists of four chapters.

The first section analyzes the terms of reference, system requirements, and possible solutions.

The second section describes the design process, analysis of the choice of software and hardware.

The third section describes the practical part of the project - creating flowcharts of algorithms, program code, and layout.

The fourth section describes life safety and the basics of labor protection.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
1.1 Аналіз вимог до комп'ютерної системи	9
1.2 Аналіз можливих вирішень поставленого завдання.....	10
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА	13
2.1 Розробка узагальненої структури комп'ютеризованої системи інформування про прийом медикаментів	13
2.2 Обґрунтування вибору апаратного забезпечення комп'ютеризованої системи інформування про прийом ліків	15
2.2.1 Arduino UNO.....	15
2.2.2 Модуль DS3231 Clock.....	20
2.2.3 Дисплей 1602 LCD	23
2.3 Обґрунтування вибору програмного забезпечення.....	28
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	32
3.1 Реалізація проєктних рішень.....	32
3.2 Тестування комп'ютерної системи.....	40
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	43
4.1. Долікарська допомога при отруєннях.....	43
4.2. Вимоги до профілактичних медичних оглядів для працівників ПК	46
ВИСНОВОК.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
Додаток А. Технічне завдання	52

					<i>КС КРБ 123.034.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Бражніков В.А.</i>			Комп'ютеризована система інформування про прийом медикаментів	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Осухівська Г.М.</i>				5		
<i>Реценз.</i>		<i>Млинко Б.Б.</i>				<i>ТНТУ, каф. КС, гр.СІ-41</i>		
<i>Н. контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Додаток Б. Перелік елементів	59
Додаток В. Лістинг програми	61

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У сучасному світі швидкого розвитку технологій та зростаючої складності медичних режимів лікування, правильний та своєчасний прийом медикаментів є вирішальним фактором для забезпечення ефективності та безпеки лікування. Проте, часто пацієнти стикаються з проблемою забуття або неправильного дозування ліків, що може негативно впливати на їх стан здоров'я та результати лікування.

З метою вирішення цієї проблеми, в рамках даної кваліфікаційної роботи була розроблена комп'ютеризована система інформування про прийом медикаментів, яка базується на платформі Arduino UNO. Arduino UNO - це мікроконтролер, який володіє широкими можливостями для створення електронних пристроїв та систем.

Основною метою цієї роботи є розробка простої, доступної та ефективної системи, яка допоможе пацієнтам забезпечити своєчасний та правильний прийом медикаментів. Система базується на використанні LCD дисплею, фізичних кнопок та модуля годинника RTC DS3231, які забезпечують збір інформації про прийом ліків та надсилають відповідні нагадування пацієнту.

Одним з ключових компонентів системи є модуль дисплея, який відображає інформацію про розклад прийому медикаментів. Крім того, система використовує звукові сигнали та світлові індикатори для додаткового інформування пацієнта.

В результаті розробки даної комп'ютеризованої системи інформування про прийом медикаментів на основі Arduino UNO, очікується поліпшення дотримання пацієнтами лікувального режиму, зменшення ймовірності забуття або неправильного дозування ліків, а також покращення загального стану здоров'я пацієнтів та результатів їх лікування.

Комп'ютеризована система інформування про прийом медикаментів може бути використана в медичній практиці для покращення управління лікувальними режимами та надання пацієнтам більш надійної підтримки при прийомі

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

медикаментів. Крім того, розроблену систему можна легко адаптувати та розширити для використання в інших сферах, де потрібен точний та своєчасний контроль за прийомом різних речовин або препаратів.

Завдяки поєднанню потужних можливостей Arduino UNO та інтеграції зі спеціалізованими компонентами, розробка комп'ютеризованої системи інформування про прийом медикаментів стає важливим кроком у напрямку покращення якості медичної допомоги та підвищення свідомості пацієнтів щодо їх власного здоров'я та лікування.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютерної системи

Комп'ютеризована система інформування про прийом медикаментів (КСІПМ) має на меті допомогти людям вчасно та точно приймати необхідні ліки. Вона може бути корисною для різних категорій користувачів, таких як: люди похилого віку, які часто мають складнощі з пам'яттю та здатністю слідувати регулярному графіку прийому ліків, людям з недостатньою пам'яттю, які можуть мати проблеми після травм мозку чи неврологічних розладів та іншим категоріям осіб, що перебувають на медикаментозному лікуванні.

Коли мова йде про лікарні, де багато хворих і складно прослідкувати за прийомом медикаментів кожного пацієнта, у справу входить фактор людини і традиційні способи нагадування можуть збоїти. У таких випадках на допомогу приходять технології і ми можемо використати електроніку. Застосування комп'ютеризованої системи про прийом медикаментів дуже широке і може використовуватися пацієнтами вдома, лікарями в лікарнях і в багатьох інших місцях.

У кваліфікаційній роботі описано процес розробки системи з оповіщення про прийом медикаментів на основі Arduino UNO. Призначенням даної системи є звукове та світлове нагадування для користувача про настання часу для прийому медикаментів. Для виконання цих задач система повинна відповідати наступним вимогам:

- Компактність: Система не повинна займати багато простору і має бути в компактному розмірі.
- Низька собівартість: Вартість системи має бути низькою, щоб була доступною для широкого кола користувачів.

					<i>КС КРБ 123.034.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Бражніков В.А.</i>			АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Осухівська Г.М.</i>					9	
<i>Реценз.</i>		<i>Млинко Б.Б.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

- Можливість вибору декількох режимів оповіщення: Система повинна мати можливість налаштування різних режимів нагадування, в залежності від вимог користувача.

- Нагадування про прийом медикаментів: Система повинна мати функціонал для нагадування користувачу про необхідність прийому медикаментів згідно заданого графіку.

- Звукові сигнали: Система повинна мати можливість відтворювати звукові сигнали для ефективного нагадування про прийом медикаментів.

1.2 Аналіз можливих вирішень поставленого завдання

У наш час популярними засобами нагадування є мобільні додатки, які надсилають сповіщення користувачу в заданий ним час. Так, компанія Apple інтегрувала дану функцію у в операційну систему свого смартфона, зовнішній вигляд додатку зображено на рис. 1.1.

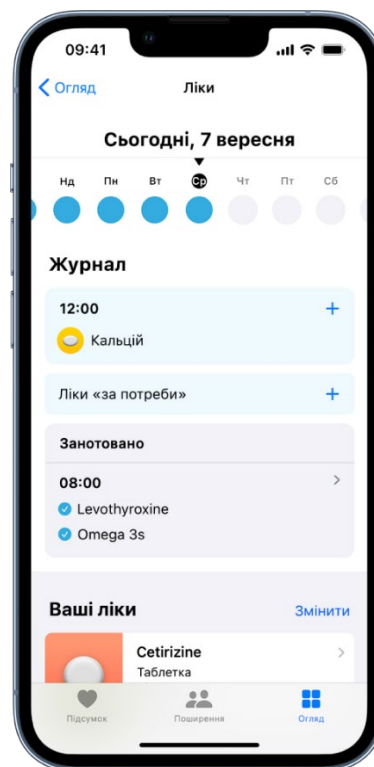


Рисунок 1.1 – Зовнішній вигляд додатку для інформування про прийом медикаментів

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Функціонал додатку є досить широким: запис речовин, які потрібно приймати, графік прийому ліків, доступ до широкої бази даних медикаментів у мережі інтернет.

Але існують також користувачі, які не мають смартфона чи не можуть в міру їх обмежених можливостей ним користуватися [1].

Щодо нагадування, існує декілька способів:

- представлення на цифровому дисплеї;
- сповіщення на електронну пошту чи мобільний телефон;
- мобільний додаток;
- налаштувати будильник;
- отримати дзвінок.

Залежно від потреб можна комбінувати способи сповіщення.

Для виконання кваліфікаційної роботи можна використати три платформи: Arduino, ESP32 або Raspberry Pi. У кожній з цих систем є власні переваги та недоліки.

Arduino орієнтована на розробку простих або середньої складності електронних пристроїв, часто може бути використана для прототипів або вбудованих систем. В плані функціоналу, вона має доволі простий інтерфейс та невелику кількість вбудованих периферійних пристроїв, що орієнтовані в основному на контролі вводу-виводу. До Arduino можна підключити велику кількість додаткових модулів, таких як сенсори, екрани, мотори, це робить дану платформу гнучкою для розширення функціоналу системи. Платформа Arduino має просту програмувальну модель та інтуїтивний інтерфейс.

Raspberry Pi, порівняно з Arduino, має більш розширений функціонал та обчислювальну потужність, може бути використана як повноцінний комп'ютер з операційною системою. В основному її використовують для об'ємних задач, наприклад веб-серверів чи мультимедійних систем. Raspberry Pi має вбудований Ethernet та USB-порти, має підтримку великої кількості операційних систем та програмних додатків. Дана платформа має велику кількість сумісних модулів, здатна підключати периферійні пристрої через USB та HDMI. Підтримує

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

розширювальні дошки та NAT-модулі. Raspberry вимагає при налаштуванні більшої кількості знань програмування та налаштування операційних систем.

ESP32 орієнтована на розробці проектів інтернету речей та бездротових додатків. З переваг – вбудований Wi-Fi та Bluetooth, що дає їй змогу взаємодіяти з іншими пристроями. Платформа ESP, як і інші, має підтримку різних сенсорів та модулів, підтримує роботу з хмарними сервісами. ESP32, для свого налаштування, вимагає високий рівень знань програмування та бездротових мереж.

Отже, для виконання проекту комп'ютеризованої системи інформування про прийом медикаментів було обрано платформу Arduino через її розширювані можливості, простоту програмування та зрозумілість інтерфейсу. КСПМ призначена для нагадування про прийом ліків 1, 2 або 3 рази на добу, відповідно до вибраного часового проміжку. Інтервал між добовими прийомами медикаментів складає 6 годин та відбувається, наприклад, зранку о 8 годині, в обідній час о 14, та ввечері о 20. Дані проміжки можна змінювати у кодї програми.

Часовий проміжок можна обрати за допомогою фізичних кнопок на макетній платі.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютеризованої системи інформування про прийом медикаментів

Для проєктування системи було проведено дослідження та обрано необхідне апаратне та програмне забезпечення. Також було створено узагальнену схему роботи системи нагадування про прийом ліків, що продемонстрована на рис. 2.1.

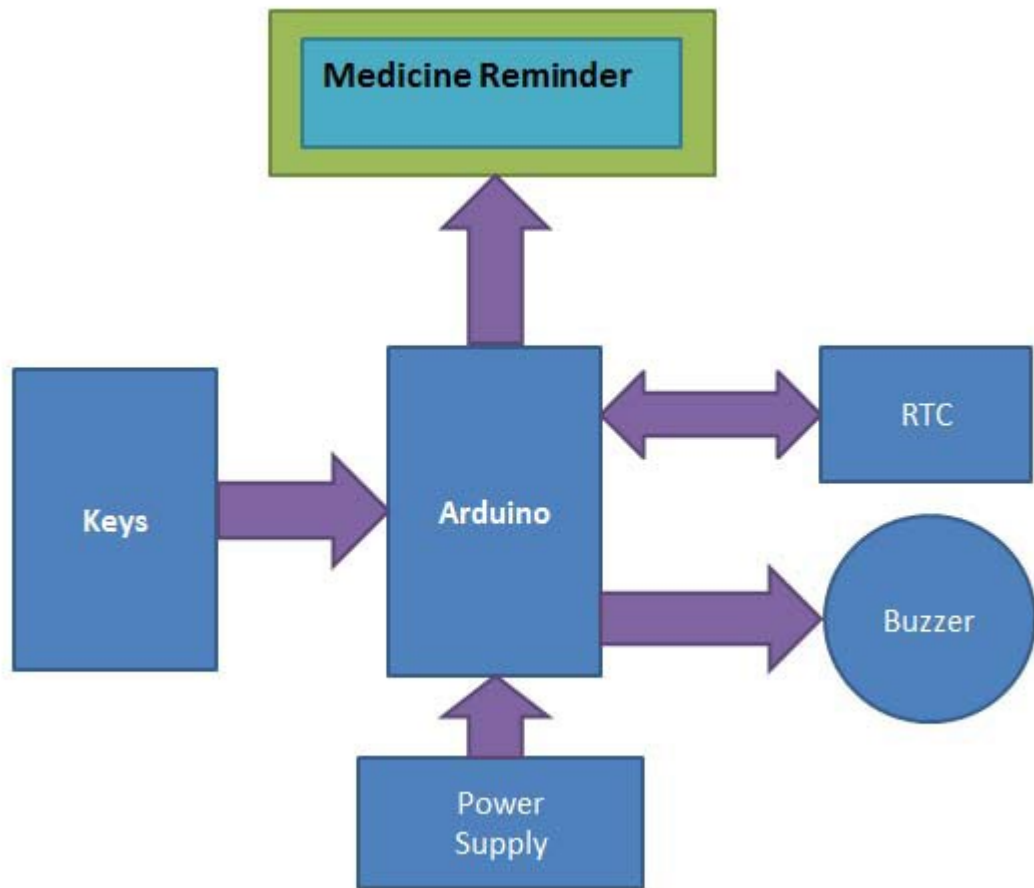


Рисунок 2.1 – Схема роботи комп'ютеризованої системи нагадування про прийом медикаментів

					<i>КС КРБ 123.034.00.00 ПЗ</i>		
Змн.	Арк.	№ докум.	Підпис	Дата	ПРОЄКТНА ЧАСТИНА ТНТУ, каф. КС, гр. СІ-41		
Розроб.		Бражніков В.А.					
Перевір.		Осухівська Г.М.					
Реценз.		Млинко Б.Б.					
Н. контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Акрушів
						13	

В основі КСІПМ використана плата мікроконтролерів Arduino UNO та модуль годинника RTC DS3231 для вимірювання інтервалів часу.

Опираючись на узагальнену схему (див. рис. 2.1) та завдання було обрано такі елементи для створення даної системи:

- Arduino UNO – плата мікроконтролерів на базі чіпу ATmega328P. Має 14 цифрових входів та виходів, з яких 6 ШІМ та 6 аналогових. Перевагою Arduino є частота її використання, відкритість та простота програмного коду.

- Модуль RTC DS3231 (Real-Time Clock) – модуль реального часу, який використовується для точного відліку в комп'ютерних системах та пристроях. Забезпечує високу точність та стабільність вимірювання.

- 16x2 Liquid Crystal Display (LCD) – дисплей для виводу тексту, що складається з 16 символних стовпчиків та 2 рядків та може відображати до 32 символів одночасно.

- Buzzer - електронний активний звуковий пристрій, який може відтворювати прості мелодії, пицання, сигнали тривоги та інші звукові ефекти.

- LED індикатор;
- Потенціометр з опором до 10кОм;
- Резистори з опором 1 та 10кОм;
- Макетна плата;
- Проводи для з'єднань.

Також була розроблено схематичний варіант підключення основних компонентів комп'ютерної системи до контролера Arduino з допомогою середовища , що зображена на рис. 2.2. Серед підключених компонентів є дисплей LCD 1602 та мікроконтролер RTC DS3231.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

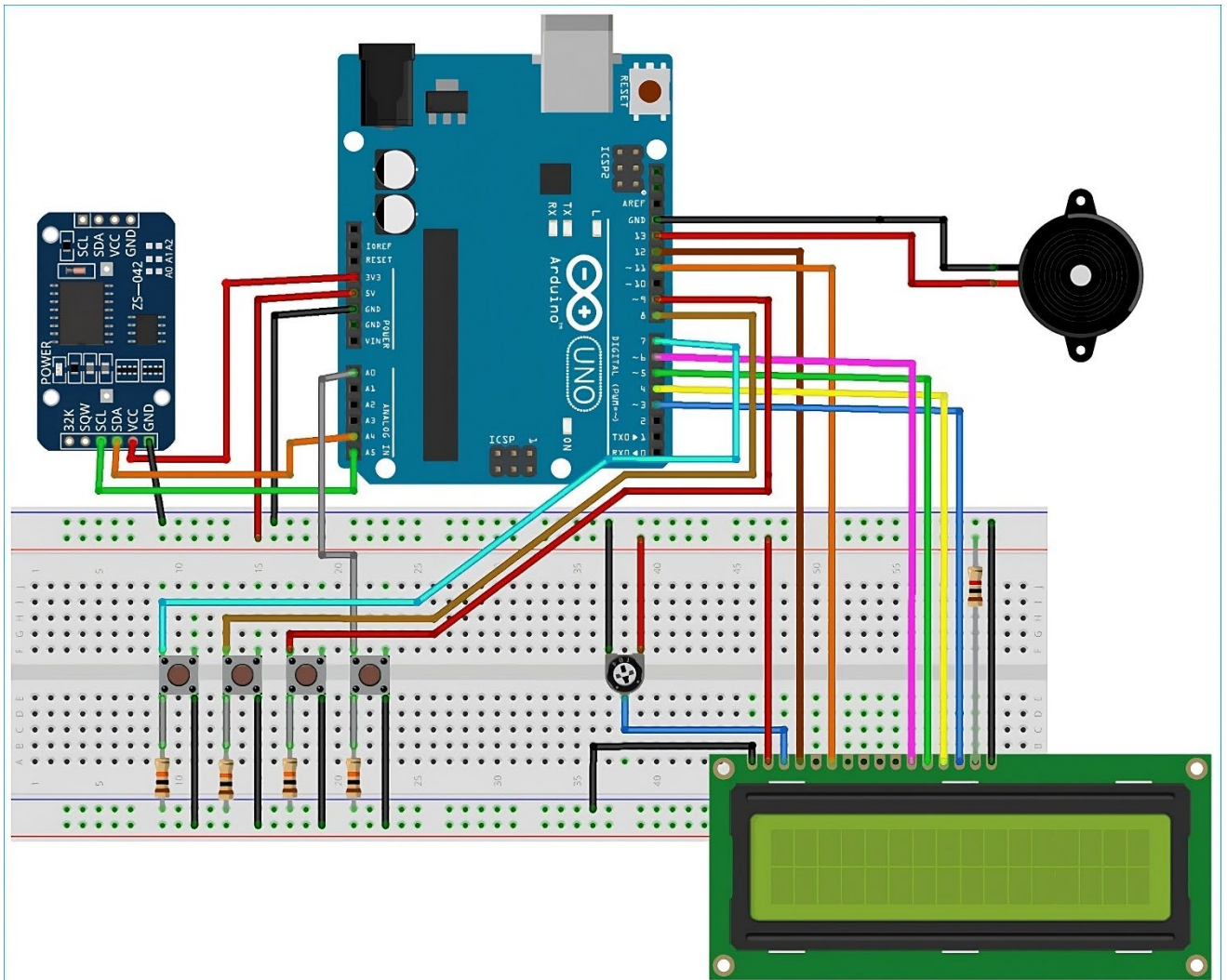


Рисунок 2.2 – Принципова електрична схема підключення основних компонентів

2.2 Обґрунтування вибору апаратного забезпечення комп'ютеризованої системи інформування про прийом ліків

2.2.1 Arduino UNO

КСІПМ виконана на основі Arduino UNO. Arduino Uno є компактним пристроєм, на основі мікроконтролеру ATmega328, який має вбудовані всі необхідні компоненти для зручної роботи. Плата налічує 14 цифрових портів, 6 з них використовуються як ШІМ-виходи для управління пристроями з регульованим сигналом. Також є шість аналогових вхідних портів, що дозволяють зчитувати аналогові значення з сенсорів або інших джерел.

									Арк.
									15
Змн.	Арк.	№ докум.	Підпис	Дата	<i>КС КРБ 123.034.00.00 ПЗ</i>				

Тактова частота вбудованого кварцового резонатора Arduino дорівнює 16 МГц, що забезпечує точну та синхронізовану роботу з мікроконтролером. Плату можна під'єднати до персонального комп'ютера через USB-порт, що дозволить завантажити та виконати код програми. Крім того, є роз'єм для живлення, який дозволяє підключати пристрій до AC/DC-адаптера або отримувати живлення від батарейки. Також присутній ICSP-роз'єм, що дає можливість програмувати мікроконтролер безпосередньо на платі [2].

Одна з відмінностей Arduino Uno від попередніх версій плат Arduino полягає у використанні мікроконтролера ATmega16U2 як перетворювача USB-UART інтерфейсів. Це замінює використання окремої мікросхеми FTDI, що спрощує процес взаємодії з комп'ютером та забезпечує підтримку широкого спектру ОС та програмного забезпечення.

Arduino Uno - це надійна та універсальна плата для розробки, яка проста у використанні та має велику спільноту користувачів. Вона ідеально підходить як для студентів так і професіоналів, які бажають створити власні проекти з використанням мікроконтролерів.

Розміри друкованої плати Arduino складають 69мм в довжину і 54мм в ширину, враховуючи виступи USB-роз'єму і роз'єму живлення за межі плати. Чотири отвори для кріплення дозволяють закріпити плату на поверхні або у корпусі. Варто зазначити, що відстань між цифровими виводами D7 і D8 не дорівнює стандартним 2.54 мм, а збільшена до 4 мм. На рис. 2.3 зображено зовнішній вигляд плати.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.3 – Зовнішній вигляд плати Arduino UNO

На плату подається живлення через USB-порт або зовнішнє джерело, його вибір відбувається автоматично. Зовнішнім джерелом живлення може слугувати AC/DC адаптер або акумуляторна батарея. Штекер діаметром 2.1мм вставляється у спеціальний роз'єм живлення на платі. У випадку прямого живлення від батареї, її дроти потрібно під'єднати до роз'ємів VIN та GND.

Для стабільної роботи усіх компонентів на платі потрібне живлення від 6 до 20В. Зменшення напруги нижче 7В призведе до нестабільної роботи компонентів, які були підключені до плати. Підвищення рівня напруги вище 12В може призвести до перегріву стабілізатора, що зробить плату непридатною для використання. Отже, найстабільніший рівень роботи Arduino UNO забезпечить напруга 7-12В.

Основні виходи живлення на платі:

- VIN (живлення) – напруга безпосередньо від зовнішнього джерела (не пов'язано з USB/5V);
- Живлення 5V – перетворює будь-яке вхідне живлення (USB, VIN, AC/DC) на вихідні 5V;
- Живлення 3V3 – живлення від стабілізатора на платі номіналом 3.3В;
- GND (від англ. ground – земля) – вивід заземлення на платі;
- IOREF – інформаційний вихід, що містить дані про робочу напругу плати Arduino.

									Арк.
									17
Змн.	Арк.	№ докум.	Підпис	Дата	<i>КС КРБ 123.034.00.00 ПЗ</i>				

Принципова електрична схема представлена на рис. 2.4.

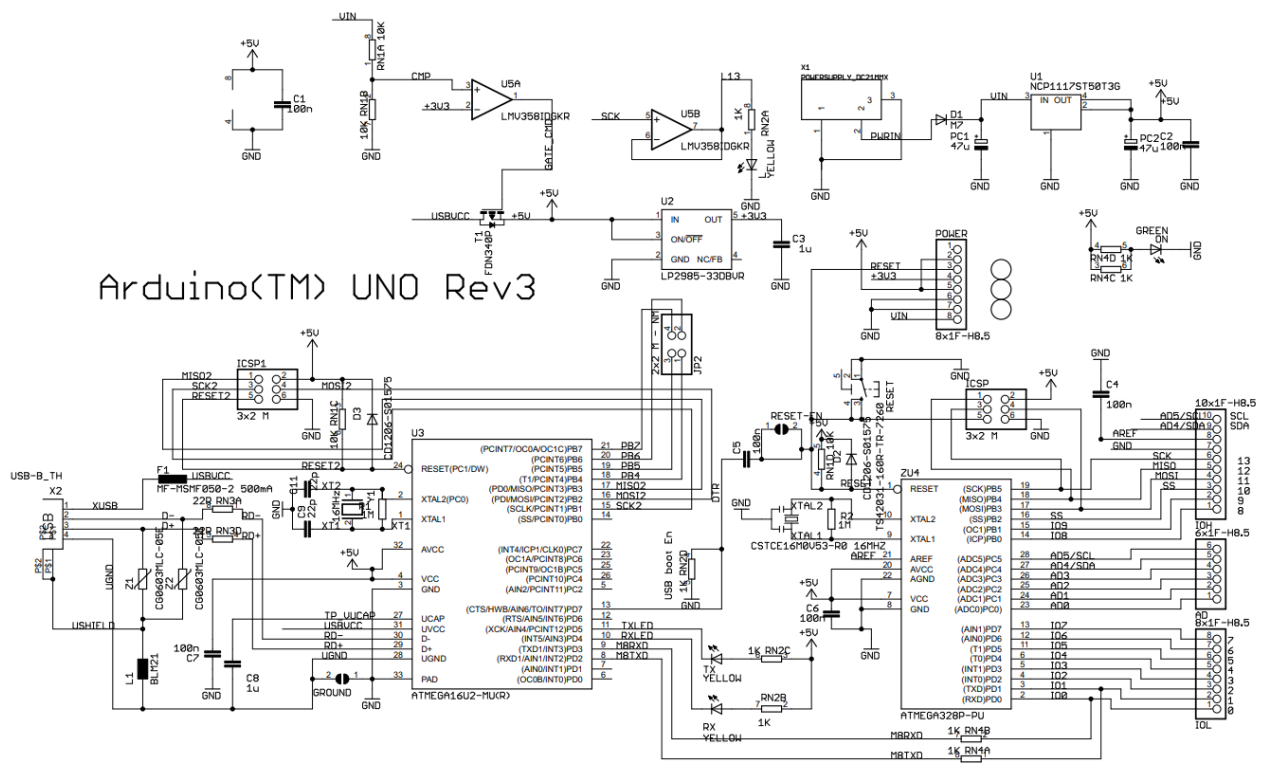


Рисунок 2.4 – Принципова електрична схема плати Arduino UNO

Використання вводу і виводу на платі Arduino виконується з допомогою програмних функцій `pinMode()`, `digitalWrite()` та `digitalRead()`. [3] З використанням цих програмних функцій кожен з 14 цифрових пінів на платі може працювати як на вхід так і вихід даних. Напруга на виходах обмежена до 5В, максимальний струм – 40мА. Окрім 14 основних роз'ємів, деякі виконують додаткові функції:

- Інтерфейси: 0 (RX) та 1 (TX) – використовуються для отримання (0) та передачі (1) даних по цифровому інтерфейсу.
- Роз'єми: 2, 3 для зовнішніх переривань - служать джерелом переривань, які виникають при підвищенні чи зниженню рівню сигналу на цих виходах.
- ШІМ роз'єми: D3, D5, D6, D9, D10, D11 – з використанням програмної функції `analogWrite()` перетворюють 8-бітні аналогові значення ШІМ-сигнал.

- SPI інтерфейси: 10(SS), 11(MOSI), 12(MISO), 13(SCK) – роз’єми, які з використанням бібліотеки SPI можуть зв’язуватися інтерфейсом SPI.
- Світлодіод: D13 –при отриманні значення програмного HIGH – вмикається, при отриманні значення LOW – затухає.
- Аналогові входи: A0, A1, A2, A3, A4, A5 – роз’єми, які перетворюють у 10-бітне число значення аналогової напруги.
- TWI: A4(SDA) або A5(SCL) – використовуються для зв’язку з TWI-інтерфейсом через програмування з бібліотекою WIRE.
- AREF – у програмі використовується функцією analogReference() як опорний вольтаж аналогових входів.
- Reset – при формуванні низької напруги на цьому роз’ємі відбудеться перезавантаження мікроконтролера.

Роз’єми плати Arduino UNO представлені на рис. 2.5.

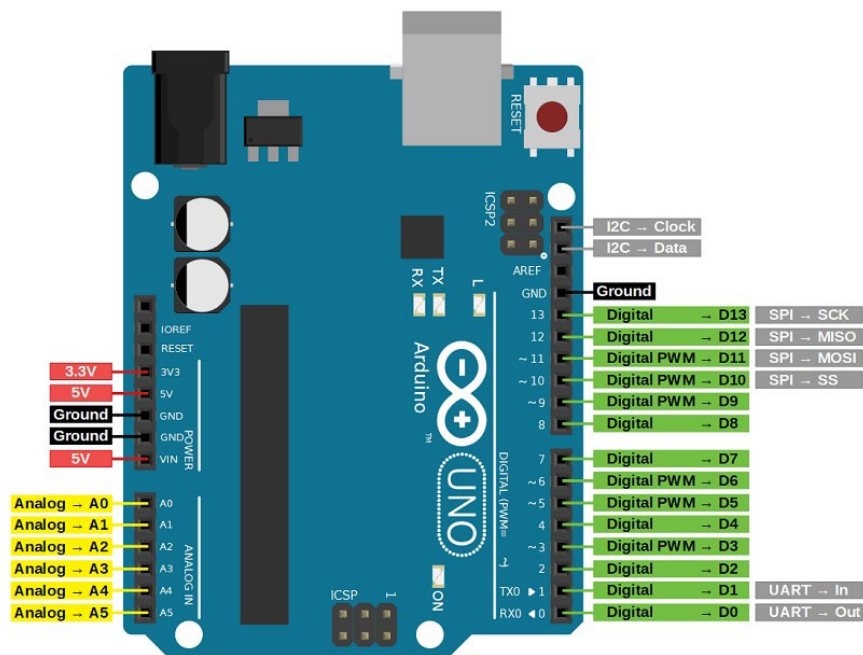


Рисунок 2.5 – Роз’єми плати Arduino UNO

Arduino Uno надає різноманітні можливості для взаємодії з комп'ютером, іншими платами або мікроконтролерами. Для послідовного зв'язку Arduino Uno використовує вбудований UART-приймач, доступний через цифрові виводи 0 (RX) і 1 (TX). Мікроконтролер ATmega16U2, що встановлений на Arduino Uno, забезпечує зв'язок цього приймача з USB-портом комп'ютера. При підключенні до комп'ютера, Arduino Uno може бути розпізнаний як віртуальний COM-порт, завдяки роботі мікросхеми 16U2 зі стандартними драйверами USB-COM. Для користування Arduino Uno під операційною системою Windows потрібний відповідний .inf-файл. У програмному пакеті Arduino є спеціальна програма, яка дозволяє передавати та отримувати прості текстові дані з Arduino Uno. Під час передачі даних через USB-UART-перетворювач, світлодіоди TX і RT на платі миготітимуть. Якщо дані передаються через виводи 0 і 1 без використання USB-перетворювача, ці світлодіоди не світяться.

За допомогою бібліотеки SoftwareSerial можна реалізувати послідовний зв'язок на цифрових виходах Arduino Uno. Мікроконтролер ATmega328 також підтримує послідовні інтерфейси I2C(TWI) і SPI. Arduino містить бібліотеку Wire, яка спрощує роботу з шиною I2C. Для отримання докладнішої інформації рекомендується звернутися до документації. Щодо роботи з інтерфейсом SPI, можна використовувати бібліотеку SPI.

2.2.2 Модуль DS3231 Clock

Модуль DS3231 володіє вбудованим TCXO і кристалом, що робить його ідеальним рішенням для економного годиннику реального часу I2C з високою точністю. Цей пристрій має вхід для батареї, що означає, що навіть при відключенні основного джерела живлення він зберігатиме точний час. Вбудований генератор забезпечує тривалу стабільність пристрою і дозволяє зменшити кількість потрібних компонентів.

DS3231 працює в широкому розкиді температур як для комерційних, так і для промислових застосувань, має компактний корпус з 16 контактами (300

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

міліметрів). Сам модуль може працювати як з системами з напругою 3,3В, так і з 5В без потреби в переключенні рівня, що робить його дуже зручним у використанні. Зовнішній вигляд модуля DS3231 та його принципову електричну схему зображено на рисунках 2.6 та 2.7 відповідно [3].



Рисунок 2.6 – DS3231 Clock Module

Далі наведено основні характеристики модуля DS3231:

- Робочий діапазон температур від -40 до +85 градусів;
- Точність синхронізації з похибкою $\pm 5\text{ppm}$ ($\pm 0,432$ секунди на день);
- Наявність резервного живлення від батареї для неперервної синхронізації;
- Економне споживання енергії;
- Пакет пристроїв та функціональні можливості сумісні з DS3231;
- Повна функціональність годинника-календаря, включаючи секунди, хвилини, години, день тижня, дату, місяць і рік, а також автоматичну корекцію високосних років до 2100 року;
- Два окремих календарних годинника;
- Вихідні сигнали з частотами 1Гц і 32,768кГц;

										Арк.
										21
Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.034.00.00 ПЗ					

- Функція скидання та попередньої обробки вхідного сигналу кнопки;
- Висока швидкість передавання даних (400кбіт/с) за допомогою послідовної шини I2C;
- Робоча напруга від +3,3 В до +5,5 В;
- Цифровий датчик температури з точністю $\pm 3\text{ }^{\circ}\text{C}$;
- Робочий температурний діапазон від -40°C до $+85^{\circ}\text{C}$;
- Компактний корпус з 16 контактами (300 млн.).

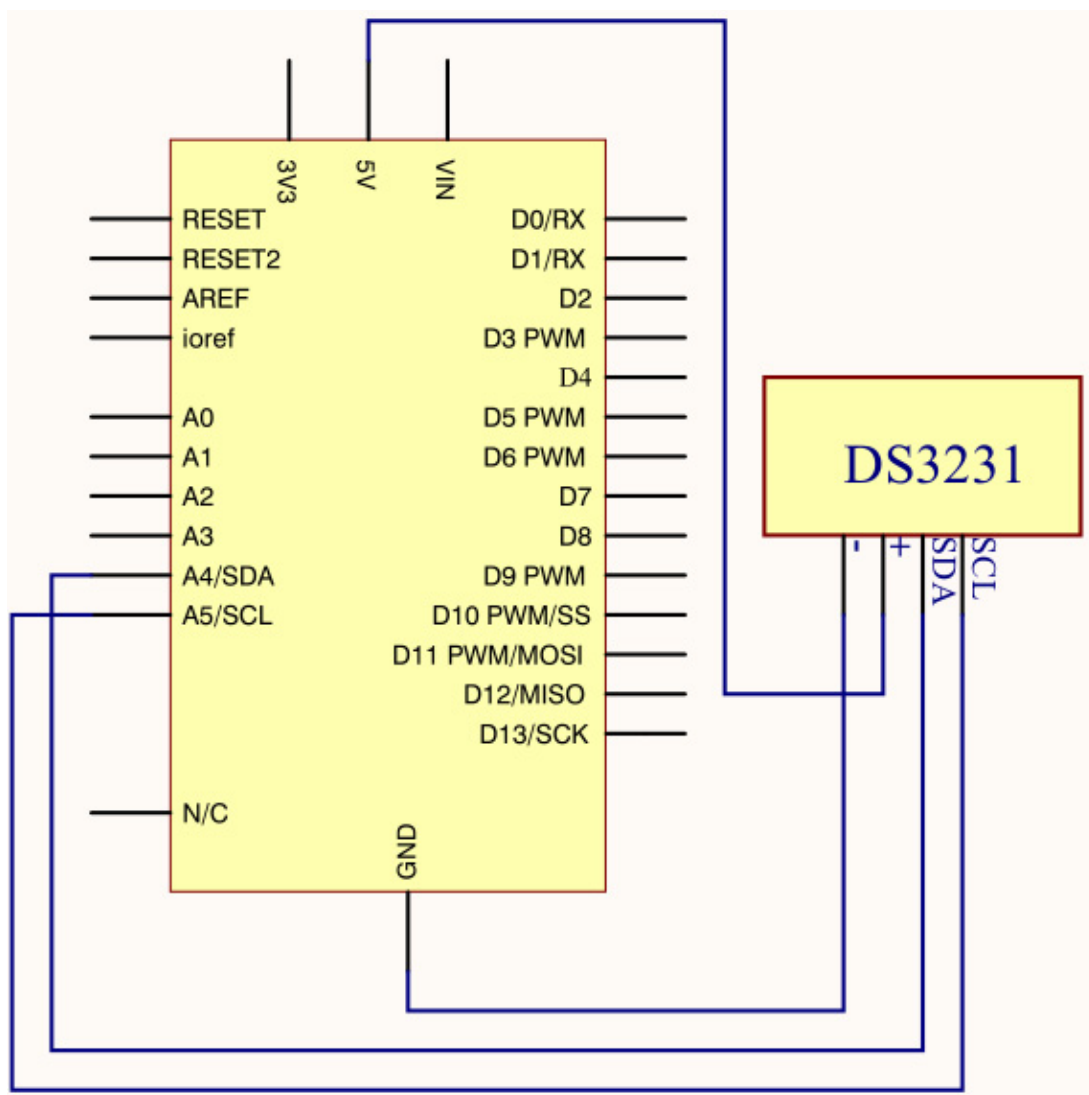


Рисунок 2.7 – Електрична схема підключення модуля DS3231

Змн.	Арк.	№ докум.	Підпис	Дата

Модуль годинника DS3231 використовує метод тестування ІС, тому для підключення модуля до Arduino потрібно приєднати “SDA” до роз’єму A4 та “SCL” до A5, живлення модуля “+” приєднуємо до VCC Arduino і – до GND Arduino. Вигляд підключення модуля годинника DS3231 наведено на рис. 2.8 [3].

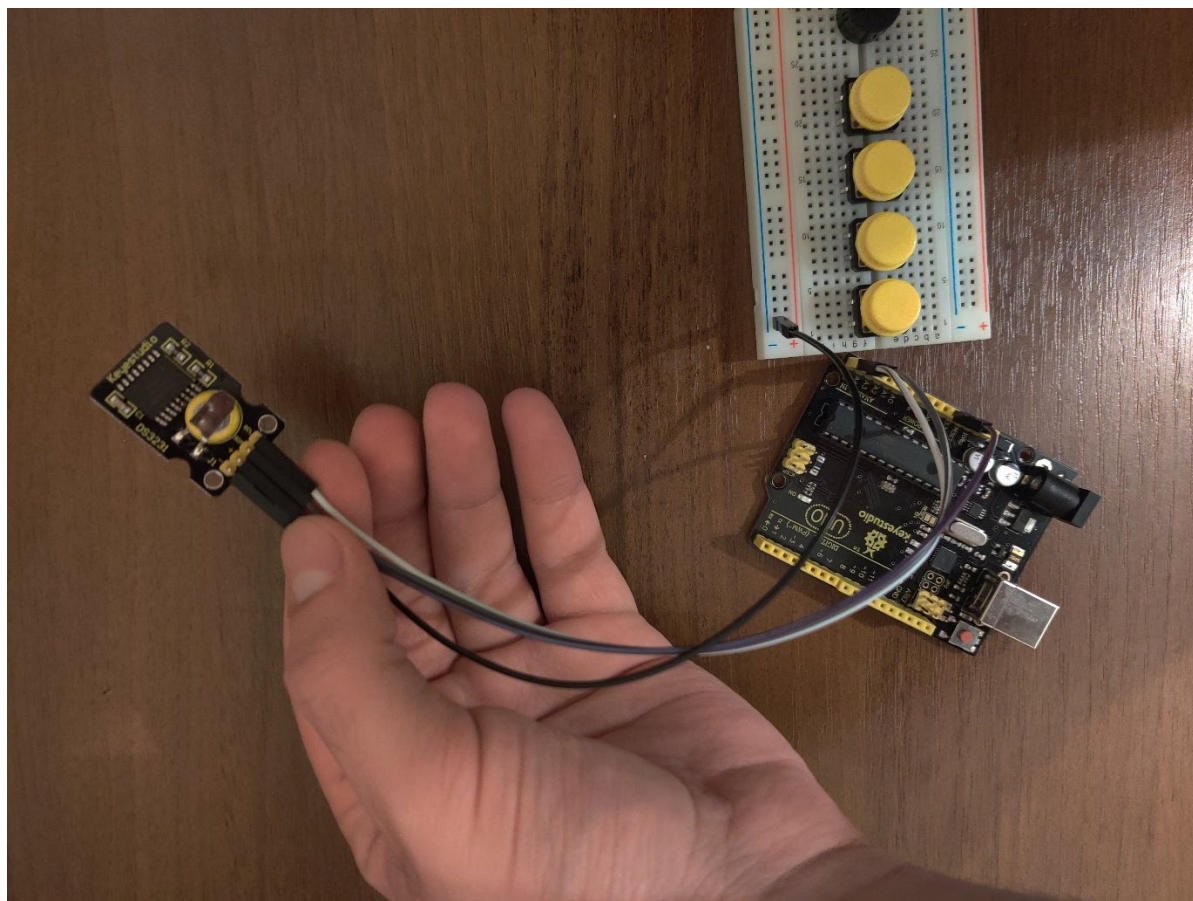


Рисунок 2.8 – Вигляд підключення модуля DS3231

2.2.3 Дисплей 1602 LCD

При виконанні кваліфікаційної роботи для виводу тексту у вигляді цифрового зображення було використано дисплей 1602 LCD. Його основними перевагами є: простота використання, порівняно невелика кількість контактів,

що дає змогу приєднати його до мікроконтролера типу Arduino UNO, зручність відображення та низька вартість. Даний дисплей використовує для роботи контролер HD44780 [4].

Основні характеристики:

- Дисплей має розмірність 16 на 2 символи;
- Мікросхема працює при напрузі 4.5 – 5.5 В;
- Робочий струм 2 мА при напрузі 5 В;
- Розмір кожного символу складає 2.95 на 4.35 мм.

Зовнішній вигляд модуля зображено на рис. 2.8.

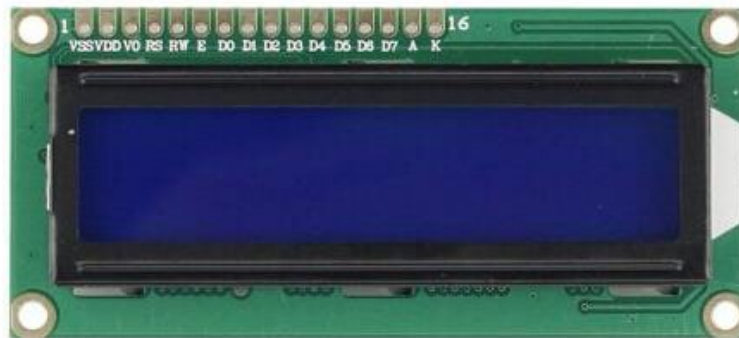


Рисунок 2.9 – Зовнішній вигляд модуля 1602 LCD

Модуль дисплею має такі основні інтерфейси:

- Два виводи для живлення: один призначений для живлення модуля, а інший - для підсвічування.
- VL - це вивід, який регулює контрастність. Зазвичай для регулювання використовується послідовно підключений потенціометр (не більше 5кОм).
- RS - це поширений вивід на РК-дисплеях, що використовується для вибору команд або даних. Коли вивід перебуває на рівні HIGH, він знаходиться в

режимі передачі даних; коли на низькому рівні, він знаходиться в режимі передачі команд.

- RW - також поширений вивід на РК-дисплеях, використовується для вибору режиму читання або запису. Коли вивід знаходиться на рівні HIGH, він знаходиться в режимі читання; якщо на низькому рівні, то він знаходиться в режимі запису.

- E - також поширений вивід на РК-дисплеях. Зазвичай, коли сигнал на шині стабілізується, він посилає позитивний імпульс, що викликає операцію читання. Коли цей вивід знаходиться на рівні HIGH, зміни на шині не допускаються.

- D0-D7 - це 8-бітна = паралельна шина, яка використовується для передачі команд і даних.

Опис контактів модуля дисплею представлено в таблиці 2.1.

Таблиця 2.1 – Опис контактів дисплейного модулю

Інтерфейс	Опис інтерфейсу
VSS	Power “-”
VDD	Power “+”
VL	Напруга (Voltage)
RS	Вибір даних/команди
R/W	Вибір запису/читання
E	Увімкнення сигналу
D0	Ввід/вивід даних
D1	Ввід/вивід даних
D2	Ввід/вивід даних
D3	Ввід/вивід даних
D4	Ввід/вивід даних
D5	Ввід/вивід даних
D6	Ввід/вивід даних
D7	Ввід/вивід даних

При роботі з 1602 LCD існує чотири базові операції: read status, write command, read data, write data. Параметри підключення та вивід відображено у таблиці 2.2.

Таблиця 2.2 – Параметри підключення LCD дисплею

Read status	Ввід	RS=L, R/W=H, E=H	Вивід	D0-D7 = status word
Write command		RS = L, R/W = H, D0-D7 = command code, E = high pulse		None
Read data	Ввід	RS = H, R/W = H, E = H	Вивід	D0-D7 = data
Write data		RS = H, R/W = L, D0-D7 = data, E = high pulse		none

Дисплей має два способи з'єднання з платою Arduino UNO – 8-біт та 4-біт. На рис. 2.9 представлено принципову схему з'єднання дисплею з платою способом 8-біт.

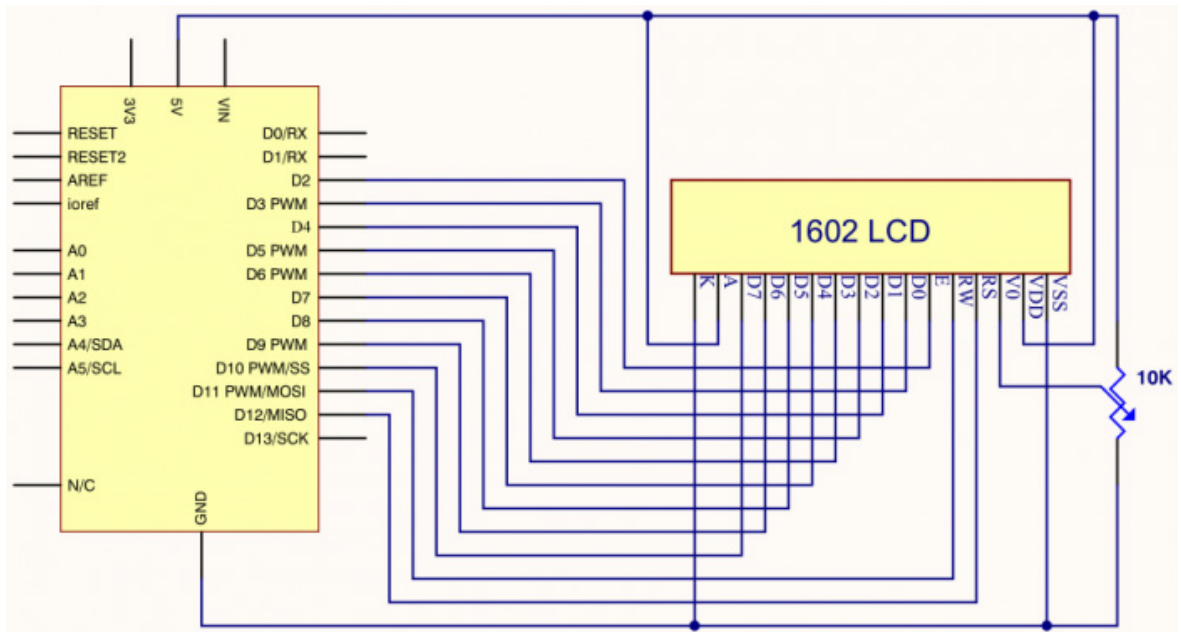


Рисунок 2.10 – Принципова схема 8-біт з'єднання дисплею з контролером Arduino

Для практичної реалізації проекту було створено макет на основі Arduino UNO та побудовано принципову електричну схему підключення основних компонентів, таких як LCD 1602 та RTC DS3231, що зображена на рис. 2.10.

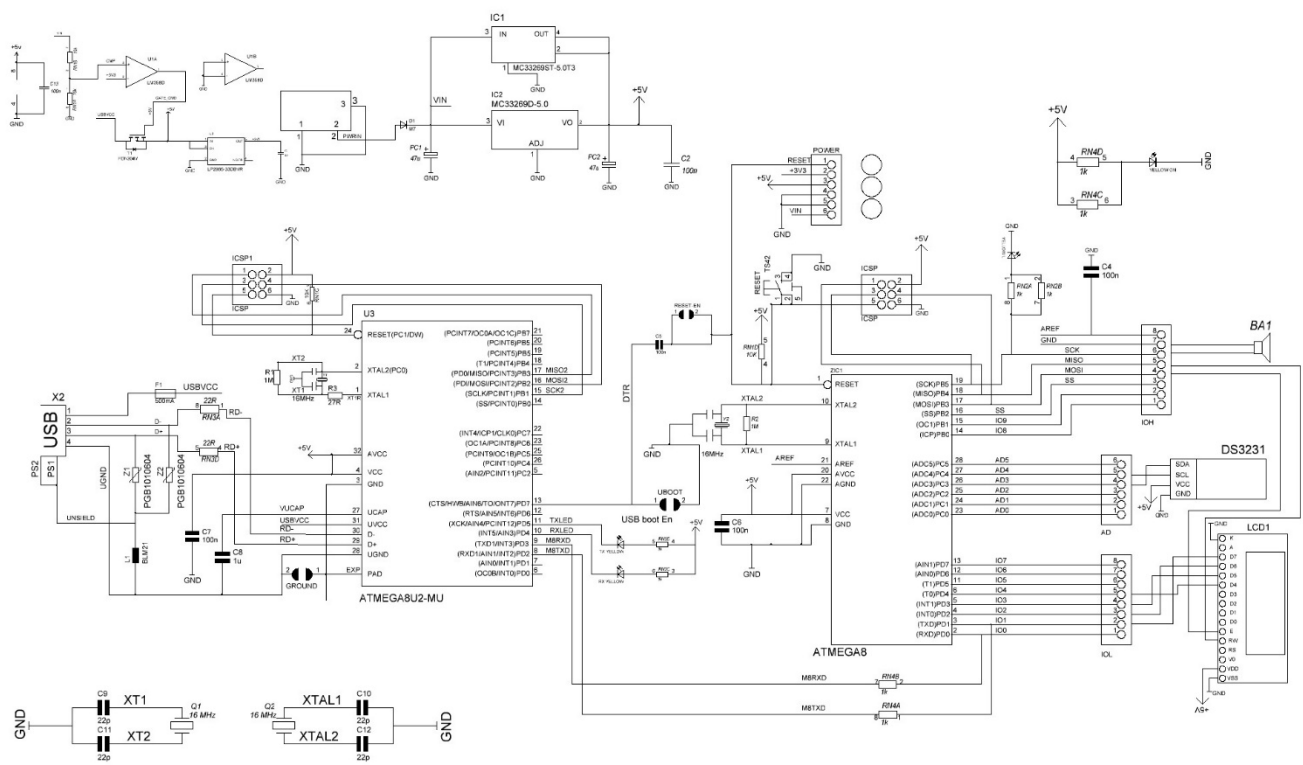


Рисунок 2.11 – Принципова електрична схема підключення компонентів до Arduino

2.3 Обґрунтування вибору програмного забезпечення

Оскільки основою системи було обрано платформу Arduino, то для розробки програми було використано програмне середовище Arduino IDE.

Arduino IDE (Вбудоване середовище розробки) - це програмне середовище, спеціально розроблене для програмування плат Arduino. Воно надає зручний та зрозумілий інтерфейс для створення, налагодження та завантаження програм на плати Arduino. Arduino IDE - це безкоштовне програмне забезпечення, яке пропонує ряд функцій для полегшення процесу розробки.

Однією з ключових особливостей Arduino IDE є вбудований редактор коду. Редактор дозволяє користувачам писати програми для Arduino мовою програмування C/C++. Він пропонує функції підсвічування синтаксису,

Змн.	Арк.	№ докум.	Підпис	Дата

автозавершення та форматування коду, що полегшує написання чистого і безпомилкового коду. Зовнішній вигляд інтерфейсу програми зображено на рис. 2.12.

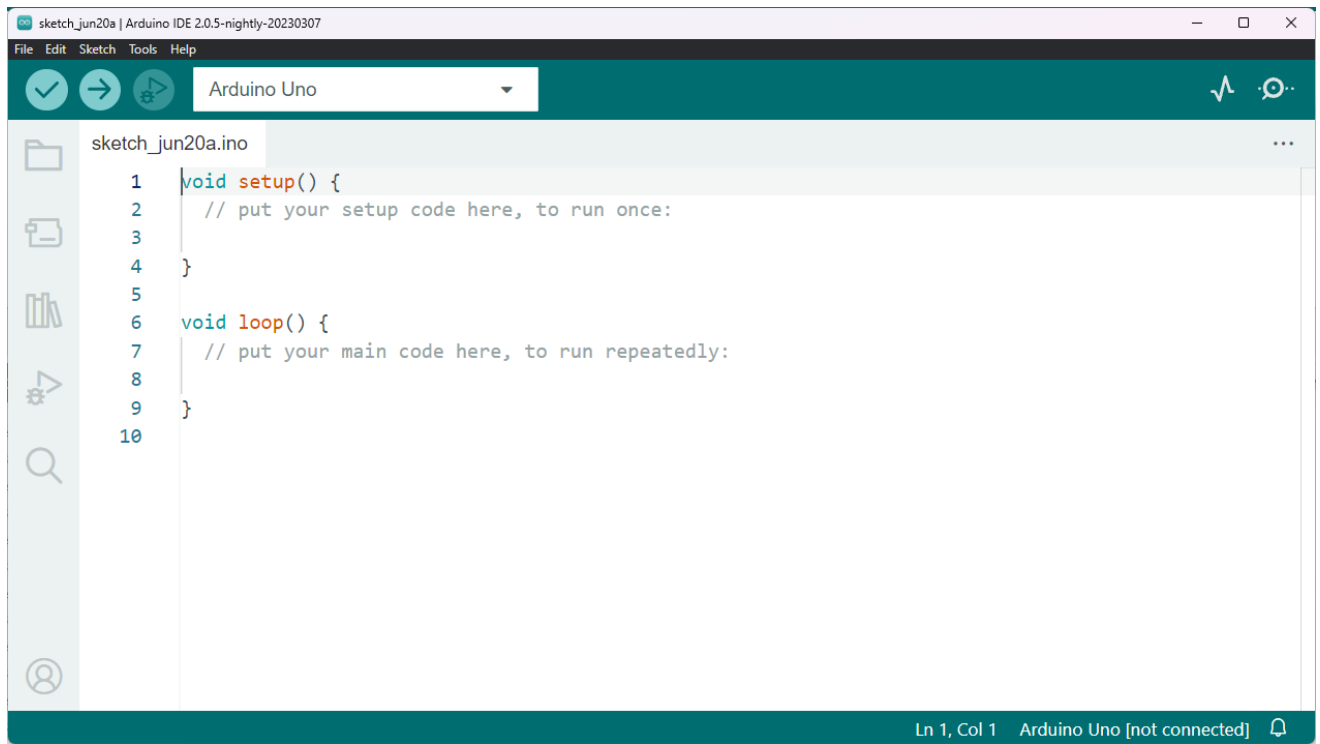


Рисунок 2.12 – Інтерфейс програми Arduino IDE

Для роботи з модулями до програми були підключені бібліотеки, щоб оптимізувати роботу з компонентами в коді, такі як: Wire.h, EEPROM, LiquidCrystal.h та RTCLib.h.

Бібліотека LiquidCrystal використовується для простого програмування LCD дисплею. Вона ініціалізує інформаційні виходи дисплею та спрощує вивід тексту на нього [4].

Бібліотека Wire.h є стандартною для середовища Arduino IDE та спрощує роботу з протоколом I2C, який використовується для встановлення зв'язку з додатковими компонентами, підключеними до плати [5].

Через наявність у комп'ютерній системі модуля RTC DS3231 до коду програми було підключено бібліотеку RTCLib.h, яка дозволяє зручно зчитати дані про час та обробити їх [6].

					КС КРБ 123.034.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Бібліотека EEPROM додає у програму функціонал для зчитування та запису даних у непостійну пам'ять, що дає змогу зберегти дані програми на мікроконтролері навіть після вимкнення живлення [7].

Arduino IDE також включає в себе інструмент компіляції та завантаження. Він дозволяє користувачам компілювати свої програми та завантажувати їх на плати Arduino через порт USB. Вибравши відповідну плату Arduino і підключений порт, користувачі можуть легко перенести свій код і протестувати його на обладнанні.

Ще однією перевагою Arduino IDE є широка підтримка бібліотек. Вона надає широкий спектр стандартних бібліотек, які забезпечують взаємодію з різними модулями та датчиками. Крім того, користувачі можуть встановлювати сторонні бібліотеки для розширення функціональності своїх програм і отримання доступу до додаткових можливостей.

Arduino IDE також пропонує вбудовані інструменти для візуалізації даних. Користувачі можуть стежити за показаннями датчиків, будувати графіки або виводити їх у вигляді тексту для аналізу з допомогою утиліти "Serial monitor". Ця функція спрощує процес збору та інтерпретації даних з підключеного обладнання. Зовнішній вигляд інтерфейсу "Serial monitor" продемонстровано на рис. 2.13.

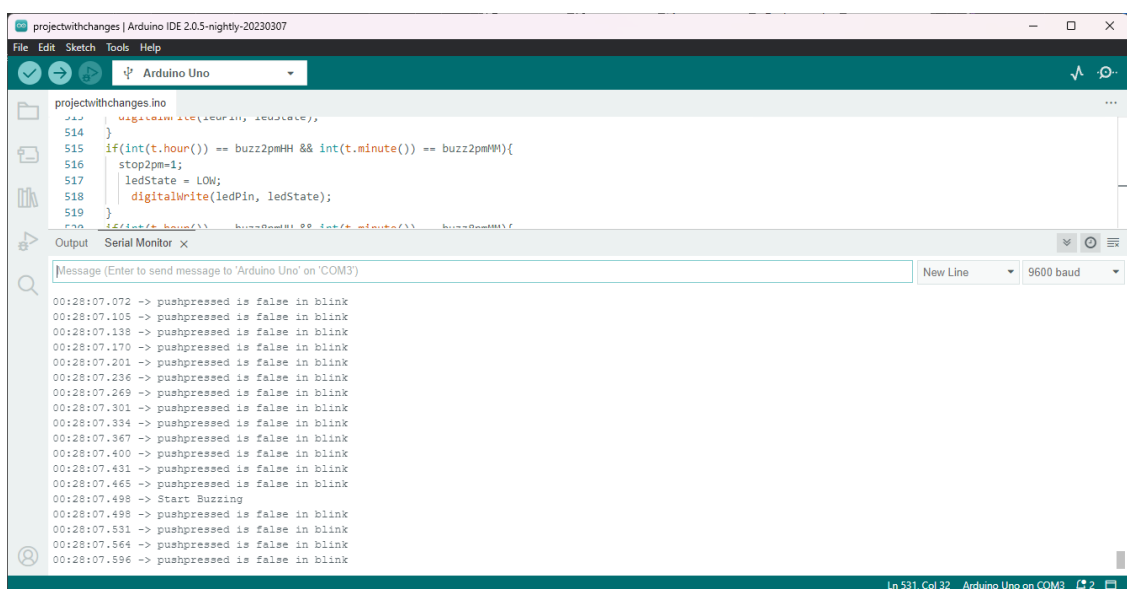


Рисунок 2.13 – Зовнішній вигляд інтерфейсу Serial monitor

									Арк.
									30
Змн.	Арк.	№ докум.	Підпис	Дата					

IDE також надає багату колекцію прикладів програм та вичерпну документацію. Ці ресурси допомагають користувачам у вивченні програмування Arduino і слугують довідниками для вирішення конкретних завдань або проблем.

Крім того, Arduino IDE підтримує кілька платформ Arduino, таких як Arduino Uno, Arduino Mega та Arduino Nano. Користувачі можуть вибрати відповідну плату зі списку і відповідно налаштувати параметри підключення.

Arduino IDE - це потужний інструмент розробки програм для плат Arduino. Зручний інтерфейс, редактор коду, можливості компіляції та завантаження, підтримка бібліотек, інструменти візуалізації даних та велика документація дають змогу працювати в ньому не лише людям з досвідом у програмуванні, а й початківцям. Arduino IDE дозволяє користувачам легко і гнучко занурюватися у світ програмування, створювати електронні проекти та взаємодіяти з фізичним світом.

Загалом, обране програмне забезпечення обґрунтовується на основі його сумісності, зручності використання, наявності необхідної функціональності, підтримки спільнотою та вартості. Враховуючи ці фактори, середовище розробки Arduino IDE, мова C++ та додаткові бібліотеки є оптимальним вибором для розробки комп'ютеризованої системи інформування про прийом медикаментів.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Реалізація проєктних рішень

Практична частина є найскладнішим та найдовшим етапом виконання кваліфікаційної роботи. Вона включає в себе реалізацію проєкту та проведення експериментальних досліджень з метою тестування.

Для функціонування системи потрібно написати програмний код, як було згадано у розділі 2, код було написано в середовищі Arduino IDE мовою програмування C++.

Програма, написана на мові C++, повинна містити ряд обов'язкових елементів, які забезпечують її коректну роботу та функціональність. Основні складові програми на C++ включають наступне:

- Підключення бібліотек: у програмі на C++ використовуються різноманітні бібліотеки для розширення функціоналу програми. Підключення бібліотек виконується на початку програми з допомогою ключового слова `#include`.
- Функція `main()` – головна функція програми, з якої починається її виконання.
- Змінні та типи даних: програма включає в себе різні типи даних такі як цілі числа (`int/float/double`), рядки (`string`) та багато інших.
- Власні функції, оголошені в коді програми.

У програмі комп'ютеризованої системи нагадування про прийом ліків було використано 4 бібліотеки: `LiquidCrystal`, `Wire`, `RTClib`, `EEPROM`.

- Бібліотека `LiquidCrystal` є популярною для програмування РК-дисплеїв (LCD) на мові C++. Вона надає набір функцій, що спрощують взаємодію з LCD-дисплеями, забезпечуючи можливість виведення

					<i>КС КРБ 123.034.00.00 ПЗ</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Бражніков В.А.			ПРАКТИЧНА ЧАСТИНА	Літ.	Арк.	Акрушів
Перевір.		Осухівська Г.М.					32	
Реценз.		Млинко Б.Б.				<i>ТНТУ, каф. КС, гр.СІ-41</i>		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

тексту, відображення символів та керування позицією тексту на дисплеї.

- Бібліотека `Wire` є стандартною бібліотекою мови програмування C++ для взаємодії з пристроями за допомогою шини I2C. Вона надає зручний і простий спосіб комунікації з пристроями, які підтримують протокол I2C, такими як сенсори, екранні модулі, EEPROM-пам'ять та інші пристрої.

- Бібліотека `RTClib` призначена для роботи з годинниковими модулями реального часу (RTC - Real-Time Clock). Ця бібліотека надає зручні функції для читання та запису дати та часу з RTC-модулів, а також для виконання різних операцій пов'язаних з часом.

- Бібліотека `EEPROM` є стандартною бібліотекою мови програмування C++, яка дозволяє зручно працювати з EEPROM - пам'яттю з можливістю електричного стирання та програмування. EEPROM використовується для збереження даних, які мають зберігатися навіть після вимкнення живлення мікроконтролера або мікропроцесора.

Перед написанням програми нам потрібно скласти алгоритм її роботи. Наша програма складається із багатьох окремих функцій, загальний алгоритм до яких складно описати. Тому надалі у вигляді алгоритму будуть описані основні функції `setup()`, `loop()`, `startbuzz()` та `changescreen()`.

Алгоритм роботи функції `setup()` зображено на рис. 3.1. Алгоритм включає в себе початок виконання програми, перевірку наявності живлення мікроконтролера RTC DS3231 та встановлення часу нагадування. Після успішної перевірки програма переходить до зчитування даних про натискання кнопки, а отже вибір кількості нагадувань на добу від користувача.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

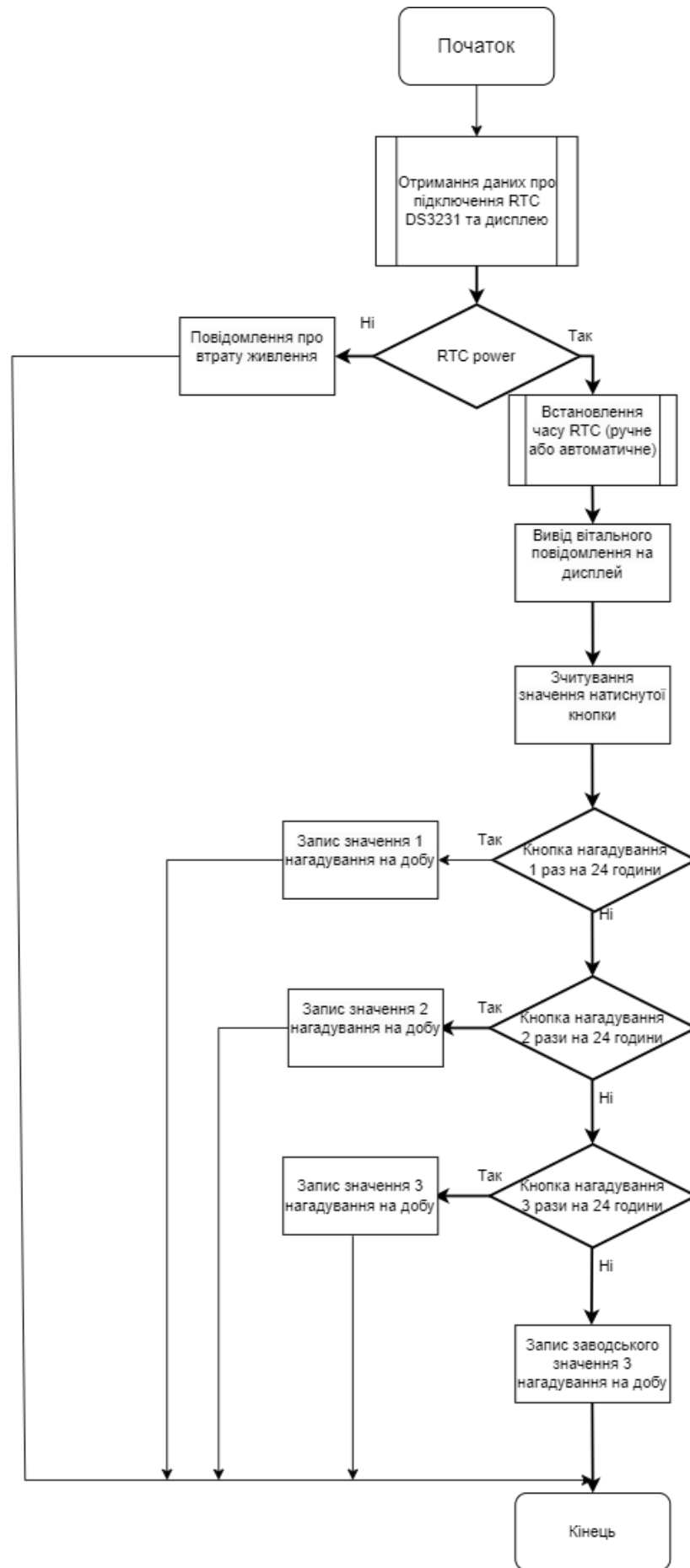


Рисунок 3.1– Алгоритм функції setup()

Змн.	Арк.	№ докум.	Підпис	Дата

Наступна функція, для якої буде написано алгоритм void loop(). Дана функція порівнює та приводить у дію значення натиснутих користувачем кнопок, а також формує дію для вимикання нагадування з використанням четвертої кнопки на платі та з допомогою функції stoppins(). Алгоритм зображено на рис. 3.2.

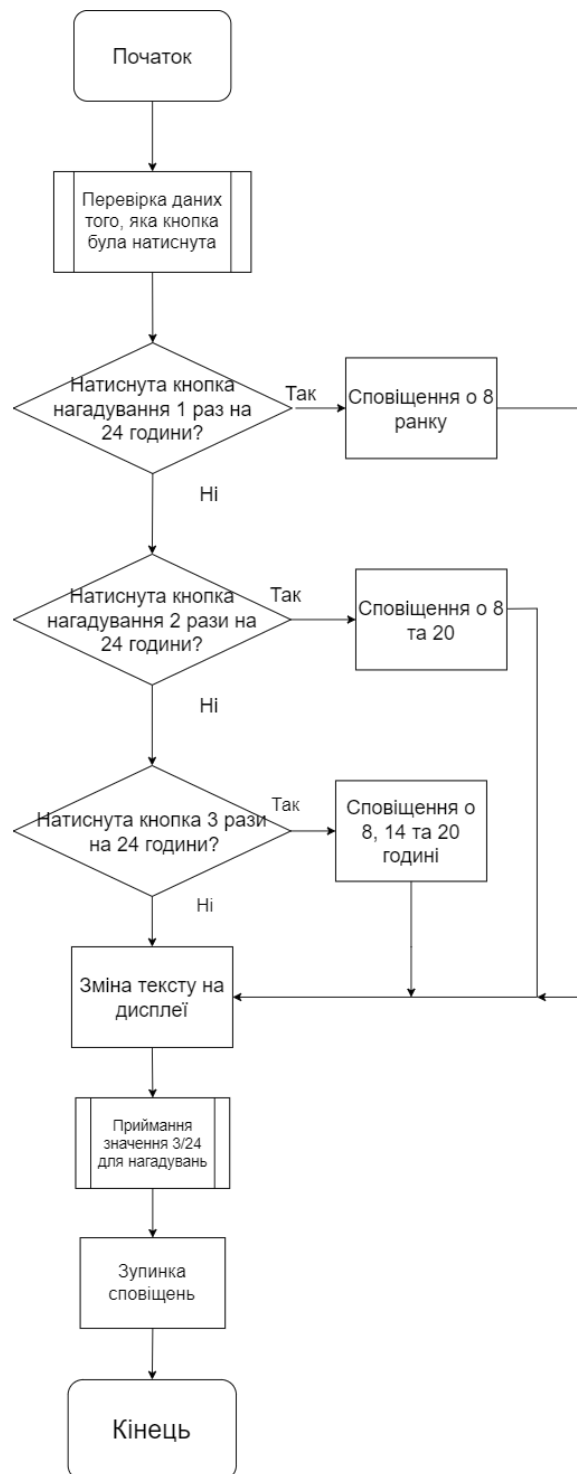


Рисунок 3.2 – Алгоритм роботи функції loop()

Функція startbuzz() проводить інформування користувача коли час на контролері DS3231 досягає потрібного значення. Інформування в програмі це початок роботи динаміка та мигання світлового індикатора. Алгоритм даної функції зображено на рис. 3.3.

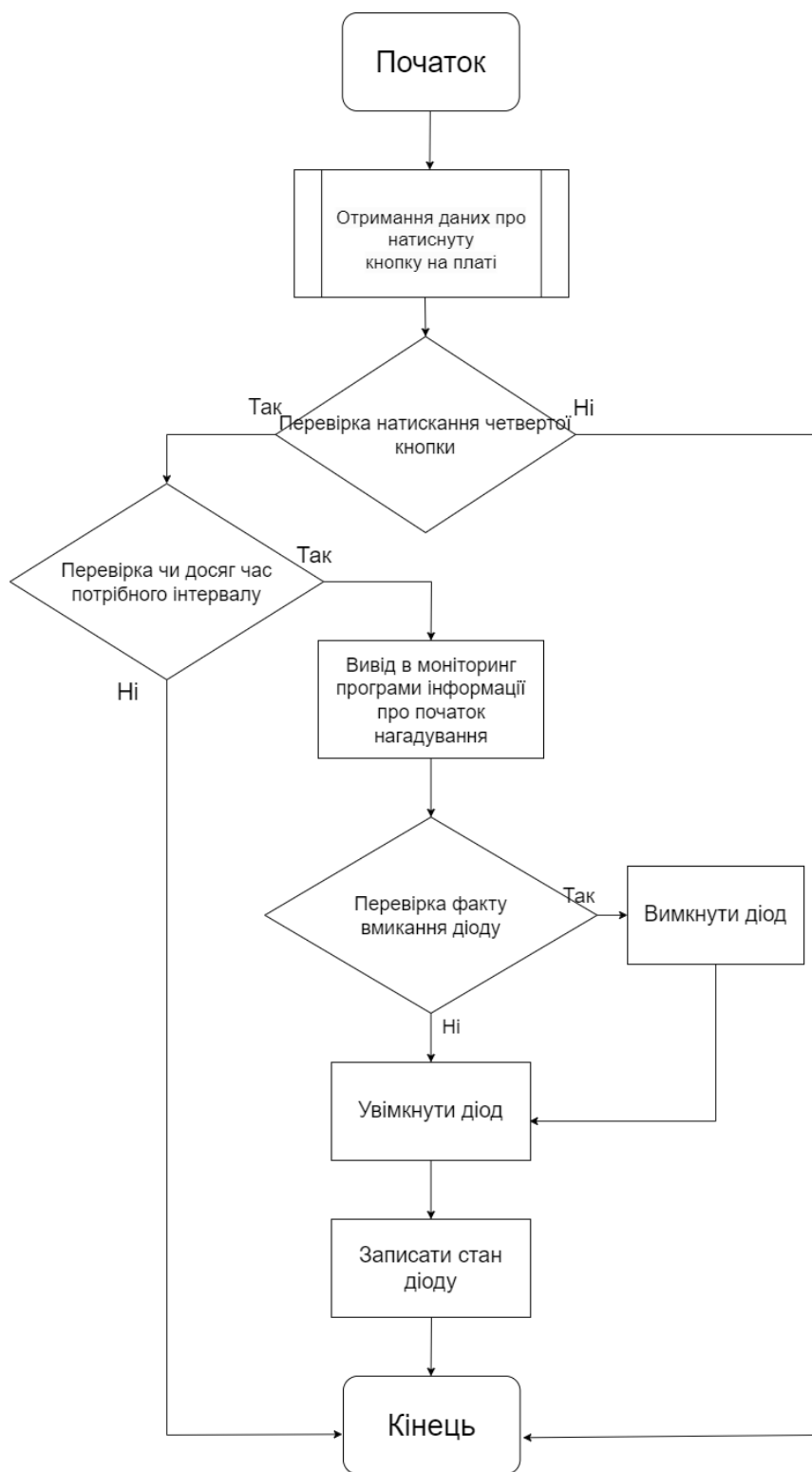


Рисунок 3.3 – Алгоритм функції startbuzz()

Змн.	Арк.	№ докум.	Підпис	Дата

Функція changescreen() призначена для інтервальної зміни тексту на LCD дисплеї. Блок-схема алгоритму функції зображена на рис. 3.4.

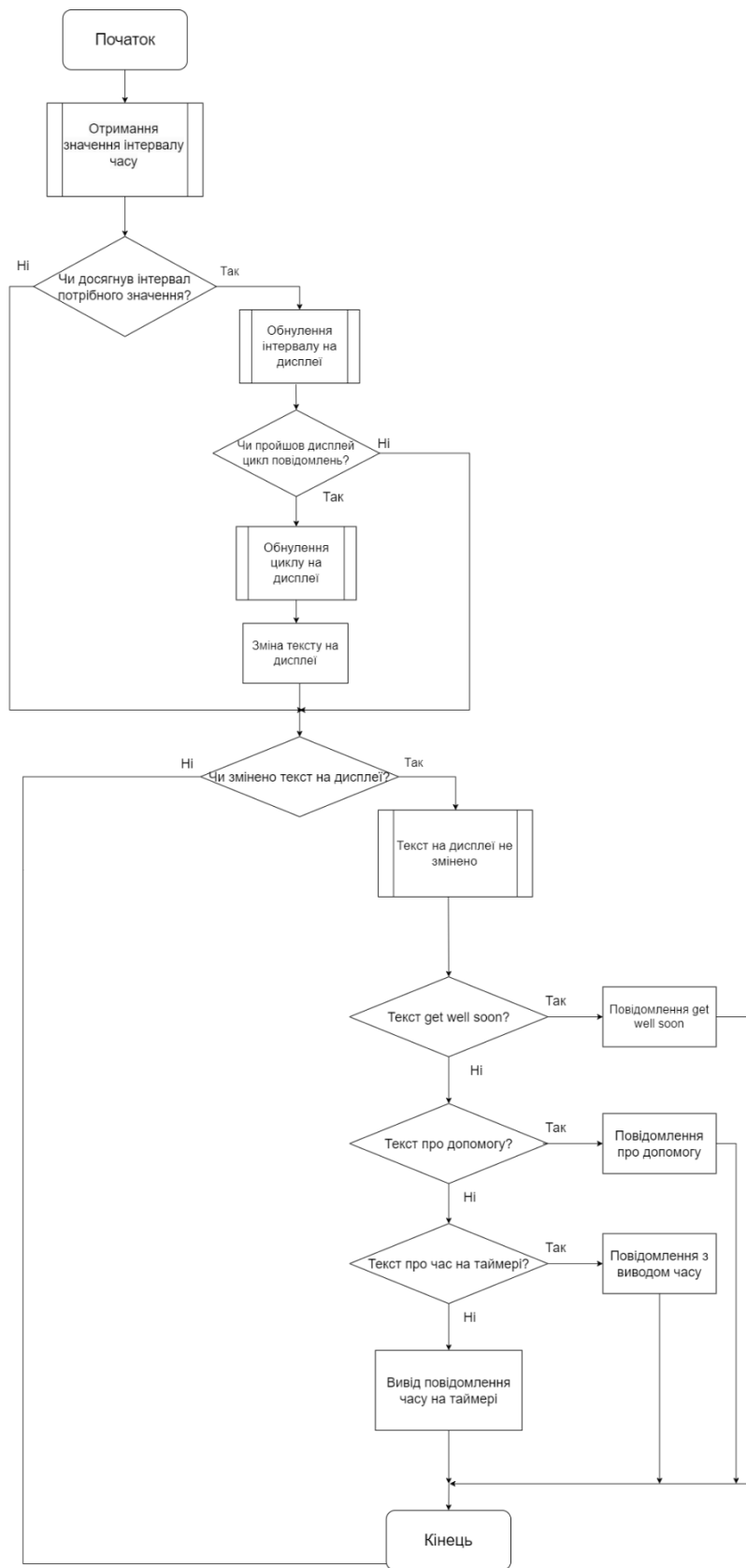


Рисунок 3.4 – Алгоритм функції changescreen()

Змн.	Арк.	№ докум.	Підпис	Дата

Для основи та підключення усіх компонентів було використано макетну плату, на якій були розміщені LCD дисплей, 4 кнопки керування, LED діод, потенціометр та активний динамік, зовнішній вигляд продемонстровано на рис. 3.5.

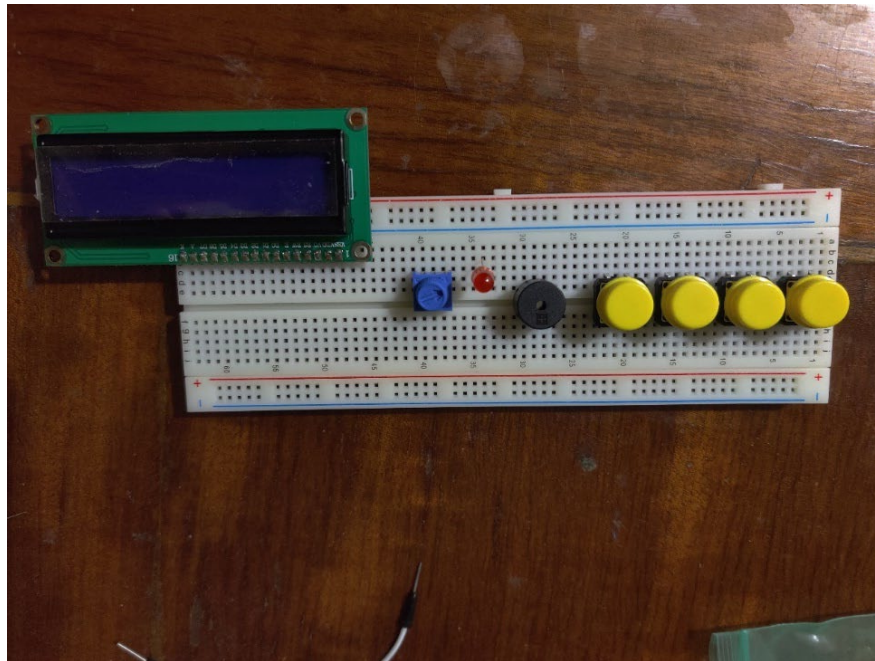


Рисунок 3.5 – Елементи на макетній платі

Наступним етапом є підключення годинника DS3231 до Arduino:

- SDA роз'єм DS3231 підключаємо до порту A4 на Arduino;
- SCL роз'єм DS3231 підключаємо до порту A5 на Arduino;
- VCC роз'єм DS3231 підключаємо до порту живлення 3.3V на Arduino;
- Роз'єм Gnd (Ground) підключаємо на відповідний порт Gnd на платі Arduino.

Підключення годинника DS3231 продемонстровано на рис. 3.6.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

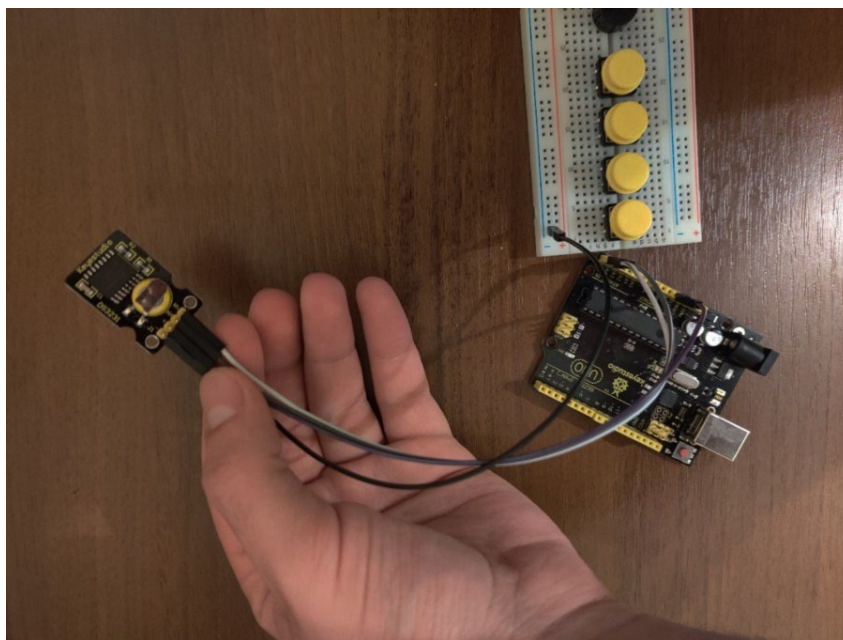


Рисунок 3.6 – Підключення контролера годинника DS3231

На рис. 3.7 зображено підключення резисторів на платі, заземлення кнопок та підключення потенціометра до LCD дисплею. В нашому випадку для підключення кнопок та дисплею на макетній платі будуть використані резистори номіналом 10кОм. Середній вихід потенціометра під'єднуємо до V0 роз'єму LCD дисплею.

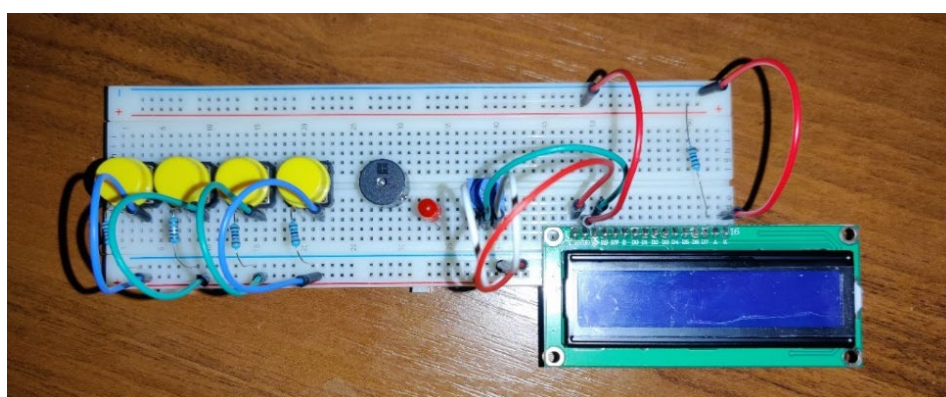


Рисунок 3.7 – Підключення дисплею та кнопок керування

Наступним етапом є з'єднання усіх компонентів на макетній платі з допомогою провідників та їх подальше підключення до Arduino UNO. До кнопок та дисплею на макетній платі під'єднано резистори з номіналом 10кОм.

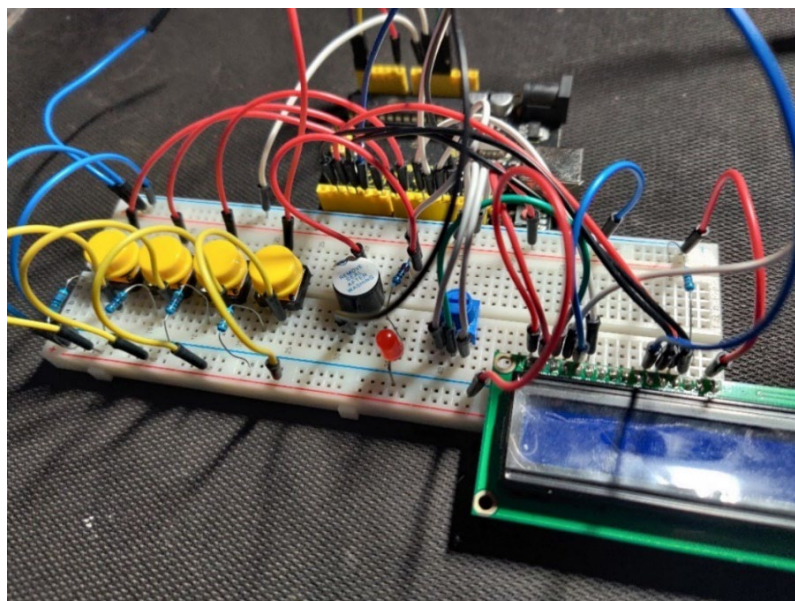


Рисунок 3.8 – З'єднання на платі з допомогою провідників

3.2 Тестування комп'ютерної системи

Після під'єднання усіх відповідних пінів дисплею, динаміка, потенціометра, кнопок та годинника DS3231 до Arduino можна розпочати тестування програми.

Для коректної роботи дисплею потрібно правильно обрати положення потенціометру на макетній платі, що показано на рис. 3.9.

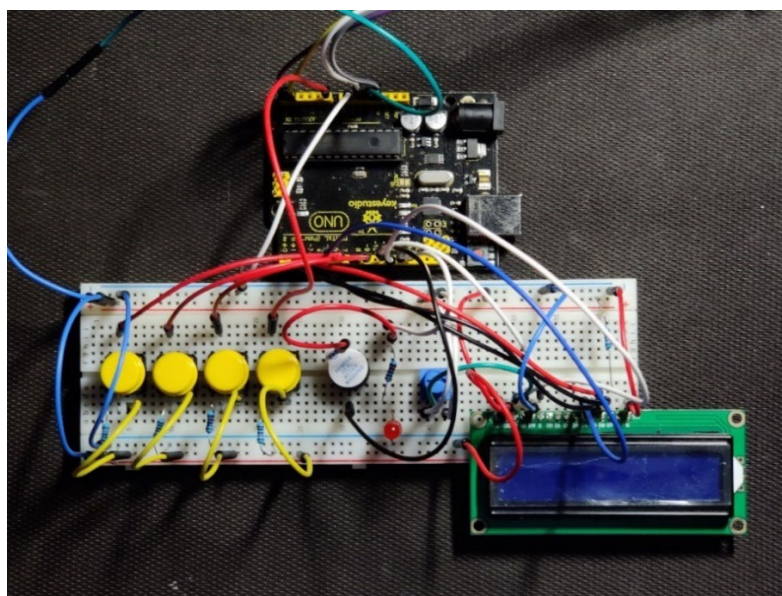


Рисунок 3.9 – Положення стрілки потенціометру на макетній платі

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Для повного тестування роботи комп'ютерної системи, вона була підключена до комп'ютера з допомогою USB-порту та перевірена на наявність підключення у середовищі Arduino IDE.

Після перевірки наявності підключення завантажуюмо програму безпосередньо на контролер з допомогою кнопки "Upload". Після успішної компіляції програми запускаємо утиліту "Serial monitor" для моніторингу роботи кнопок та датчиків. Процес компіляції та наявність підключення в середовищі програмування Arduino IDE зображено на рис. 3.10.

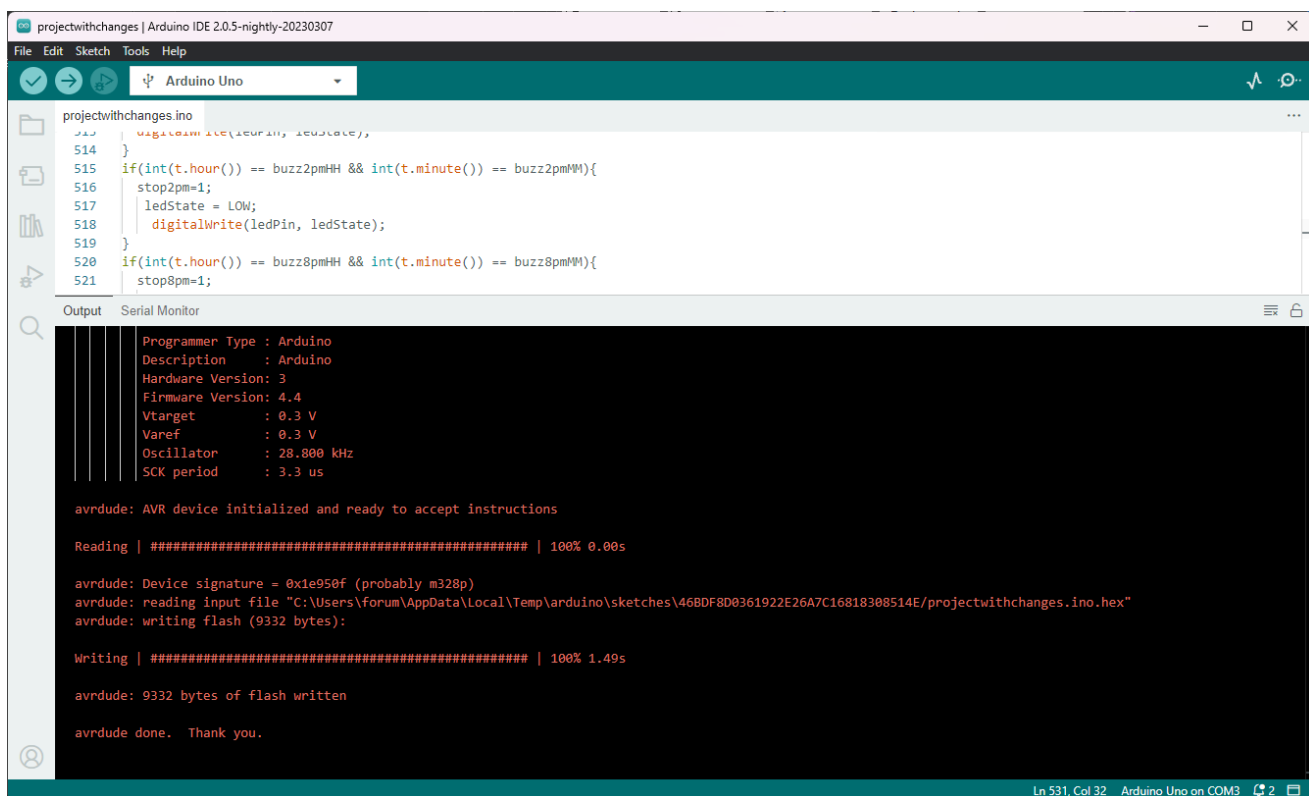


Рисунок 3.10 – Компіляція програми та наявність підключення плати Arduino

Для зручності тестування обираємо нагадування на 3 рази на добу та вручну налаштовуємо час на годиннику RTC.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

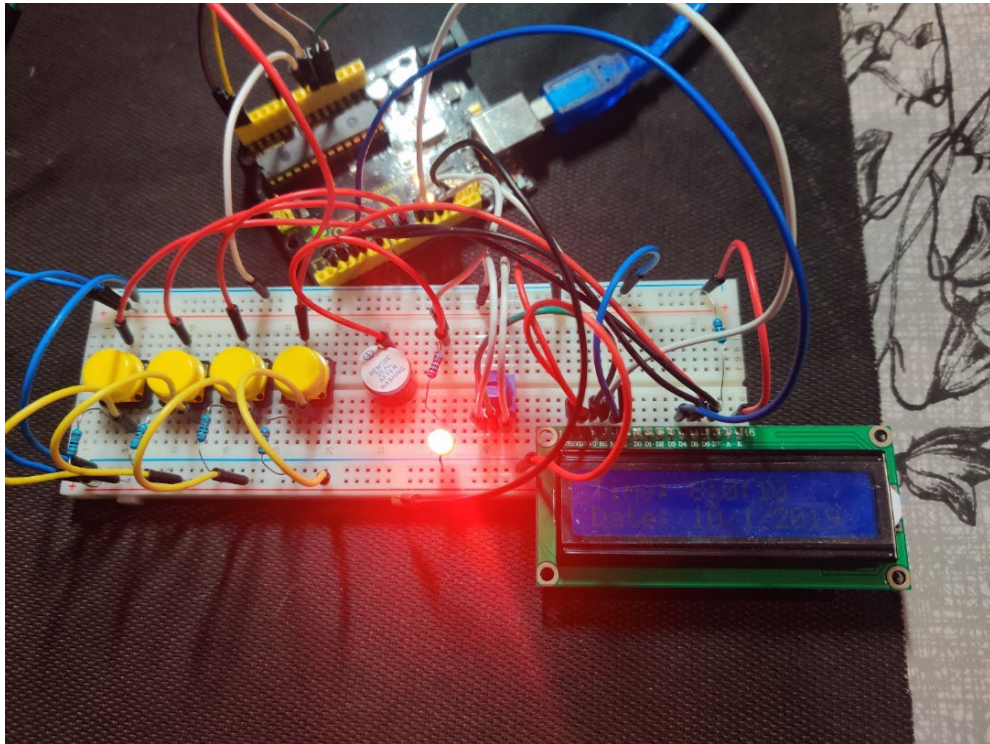


Рисунок 3.11 – Індикація сповіщення про прийом медикаментів

Після настання часу сповіщення LED індикатор починає мигати, а динамік видає звукове сповіщення, що продемонстровано на рис. 3.11.

В результаті тестування КСПМ працює коректно: після настання потрібного для сповіщення часу система сповіщає користувача про прийом медикаментів, динамік починає пищати а світловий індикатор мигати(див. рис. 3.11).

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Долікарська допомога при отруєннях.

Отруєння є серйозною загрозою для здоров'я та життя людей, і вимагає негайної та кваліфікованої медичної допомоги. Долікарська допомога при отруєннях є невід'ємною частиною системи охорони здоров'я, яка має на меті нейтралізацію токсичних речовин, відновлення функцій організму та запобігання ускладненням. Цей підрозділ кваліфікаційної роботи розгляне важливі аспекти долікарської допомоги при отруєннях, включаючи принципи та процес надання цієї допомоги.

Інтоксикації, які пов'язані із споживанням їжі, інфікованої певними видами мікроорганізмів (харчові токсикоінфекції) та речовинами різного походження, що містять токсини (власне харчові отруєння, харчові інтоксикації), є поширеними проблемами в світі. Ці захворювання мають спільний зв'язок з харчуванням та обмеженість розповсюдження іншими шляхами передачі. Отруєння бактеріальної природи становлять близько 90% загальної кількості харчових отруєнь у світі. Інфекційні харчові отруєння можуть стати наслідком споживання зараженої їжі, яка містить патогенні бактерії, такі як прострочені продукти або недотримання правил гігієни під час приготування та споживання їжі.

Характерними симптомами харчового отруєння є нудота, блювання, загальне нездужання, слабкість, лихоманка, підвищена температура тіла та діарея. Перші ознаки зазвичай з'являються протягом 2 годин після вживання непридатного продукту. У разі отруєння необхідно надати першу допомогу, яка включає промивання шлунку слабким розчином соди, чистої води або

					<i>КС КРБ 123.034.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Бражніков В.А.</i>			БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Осухівська Г.М.</i>					43	
<i>Консультант</i>		<i>Пилипець М.І.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

застосування сорбентів (наприклад, активованого вугілля), постільний режим, грілки на живіт та голодування, а також вживання багато рідин, таких як чай, розведених фруктових та ягідних соків, морси та мінеральні води.

Долікарська допомога при отруєннях має вирішальне значення для врятування життя та запобігання ускладненням. Швидка реакція та компетентна медична допомога можуть значно знизити ризик смерті та коморбідності, пов'язані з отруєннями. Вона вимагає високого рівня кваліфікації медичного персоналу, обладнання та наявності необхідних лікарських препаратів.

Долікарська допомога при отруєннях ґрунтується на кількох принципах, які допомагають забезпечити ефективність та безпеку надання медичної допомоги. Основні принципи включають:

Рання діагностика отруєння та негайне реагування є вирішальними для забезпечення успішного лікування. Медичний персонал повинен швидко впізнати ознаки отруєння, провести необхідні дослідження та ініціювати негайні медичні заходи. Важливо проводити дослідження методом анамнезу, з'ясувати можливі джерела отруєння та оцінювати ступінь токсичності речовини.

Одним із основних завдань долікарської допомоги при отруєннях є нейтралізація токсинів та їх виведення з організму. Це може бути досягнуто за допомогою застосування протидіючих агентів, активованого вугілля, специфічних протиметалевих агентів або методів промивання шлунку та кишечника. Ефективна нейтралізація токсинів допомагає знизити їх концентрацію в організмі та запобігає подальшому ушкодженню тканин та органів.

Після нейтралізації токсинів необхідно відновити функції органів, які можуть бути пошкоджені внаслідок отруєння. Це включає підтримку дихання, кровообігу та нормалізацію роботи нирок та печінки. Медичний персонал повинен використовувати відповідні методи та лікарські препарати для забезпечення належного функціонування органів.

Кожне отруєння є унікальним, тому важливо мати індивідуальний підхід до кожного пацієнта. Враховуючи вік, стать, стан здоров'я та вид отруєння, медичний персонал встановлює оптимальну схему лікування та забезпечує

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

належний контроль над станом пацієнта. Це вимагає високого рівня професійної підготовки та постійного професійного розвитку медичних працівників.

Процес надання долікарської допомоги при отруєннях може бути поділений на декілька етапів:

При першому контакті з постраждалим необхідно оцінити його стан. Важливо визначити важкість отруєння, причину та вид отруйної речовини. Медичний персонал повинен виявити наявність небезпеки для життя пацієнта та вжити необхідні заходи для забезпечення безпеки навколишніх людей.

Після оцінки ситуації медичний персонал проводить необхідні діагностичні процедури, щоб підтвердити отруєння та визначити вид токсичної речовини. Після цього ініціюється нейтралізація токсинів шляхом застосування відповідних протидіючих препаратів, активного вугілля, промивання шлунку або інших методів.

Після нейтралізації токсинів медичний персонал зосереджується на відновленні функцій організму та запобіганні ускладнень. Це може включати підтримку дихання, кровообігу, нормалізацію роботи нирок та печінки, лікування симптомів та лікарську терапію для забезпечення повного одужання пацієнта.

Після надання долікарської допомоги, медичний персонал забезпечує постійний моніторинг стану пацієнта, щоб виявити можливі ускладнення та забезпечити належний лікувальний догляд. Це може включати проведення додаткових обстежень, консультування з іншими спеціалістами, надання рекомендацій для подальшого відновлення здоров'я, які можуть включати курс прийому медикаментів, для підтримки якого можна використати КСІПМ.

Для профілактики харчових отруєнь необхідно дотримуватись певних заходів. Перш за все, важливо мити руки перед приготуванням та споживанням їжі, а також ретельно мити обробні дошки, ножі та посуд. Важливо використовувати свіжі продукти та дотримуватись технології приготування страв. Для попередження інфекційного харчового отруєння рекомендується мочити овочі, зелень та фрукти у розчині соди протягом певного часу перед

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

вживанням. Важливо не купувати продукти з пошкодженою упаковкою або прострочені продукти, а також не кладіть сире та приготоване м'ясо на одну тарілку. Зберігання продуктів при відповідній температурі та уникання вживання харчових продуктів, які можуть бути потенційно небезпечними, також відіграють важливу роль у профілактиці харчових отруєнь.

Долікарська допомога при отруєннях є важливим компонентом системи охорони здоров'я, яка має на меті врятування життя та запобігання ускладненням. Швидка діагностика, нейтралізація токсинів, відновлення функцій організму та індивідуальний підхід до пацієнта є ключовими аспектами долікарської допомоги при отруєннях. Забезпечення належного рівня кваліфікації медичного персоналу, використання сучасних технологій та постійний професійний розвиток є важливими чинниками успішного надання долікарської допомоги при отруєннях.

Отже, знання про розповсюдженість та причини харчових отруєнь, а також правильні дії у разі отруєння і профілактичні заходи можуть значно знизити ризик виникнення цих захворювань. Належна увага до гігієни під час обробки та приготування їжі, використання свіжих продуктів та дотримання правил зберігання можуть допомогти уникнути харчових отруєнь і зберегти здоров'я працівників та громади в цілому.

4.2. Вимоги до профілактичних медичних оглядів для працівників ПК

Під час використання комп'ютеризованої системи інформування про прийом медикаментів важливо дотримуватися вимог до профілактичних медичних оглядів.

Працюючі з ВДТ підлягають обов'язковим медичним оглядам: попереднім – при влаштуванні на роботу і періодичним – протягом трудової діяльності, відповідно до наказу МОЗ України N 45 від 31.03.94 р.

Періодичні методичні огляди мають проводитися раз на два роки комісією в складі терапевта, невропатолога та офтальмолога.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

До складу комісії, що проводить попередні та періодичні медичні огляди, при необхідності (за наявністю медичних показань), можуть залучатись до оглядів лікарі інших спеціальностей.

Зорова функція є одним з основних аспектів, який потребує оцінки під час медичних оглядів для працівників ПК. Це включає перевірку гостроти зору, оцінку пристосування очей до роботи за комп'ютером, виявлення ознак напруження очей, сухості та інших офтальмологічних проблем. Можуть використовуватись різні методи, такі як вимірювання зорової гостроти, скринінг на відстань близької фокусування та оцінка стану ока.

Жінки, що працюють з ВДТ, обов'язково оглядаються акушером-гінекологом один раз на два роки.

Жінки з часу встановлення вагітності та в період годування дитини грудьми до виконання всіх робіт, пов'язаних з використанням ВДТ, не допускаються.

Виконання вимог в комплексі з практичним здійсненням первинних та спеціальних заходів повинно стати нормою діяльності всіх фахівців, безпосередньо пов'язаних з навчальними та виробничими колективами.

Профілактичні огляди також мають включати загальну оцінку стану здоров'я працівників, таку як перевірка артеріального тиску, пульсу, стану шкіри та інших фізіологічних параметрів. Це може допомогти виявити можливі ознаки стресу, перевантаження, погіршення роботи серця та інших проблем, які можуть бути пов'язані з роботою за комп'ютером.

Профілактичні медичні огляди для працівників ПК можуть залежати від конкретних вимог законодавства вашої країни або регіону. Однак, незалежно від цього, існують загальні рекомендації щодо профілактичних оглядів для працівників, які проводять тривалий час за роботою на ПК. Ось деякі з них:

Офтальмологічний огляд: працівники, які проводять багато часу перед монітором, повинні регулярно перевіряти зір у офтальмолога. Це допоможе виявити будь-які проблеми із зором, такі як втома очей, сухість, роздратування та інші окулярні захворювання.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Перевірка позиції тіла: довготривала робота за комп'ютером може викликати проблеми зі статевими органами, шиєю та кистями. Працівники повинні отримувати консультацію фахівця, який допоможе налаштувати правильну позицію тіла під час роботи.

Паузи та фізичні вправи: необхідно регулярно робити паузи під час роботи за ПК та виконувати прості фізичні вправи для розслаблення м'язів і покращення кровообігу.

Психологічний стан: працівники, які працюють за ПК, можуть відчувати стрес і психологічне напруження. Потрібно забезпечити психологічну підтримку та доступ до консультанта з психологічного здоров'я для вирішення проблем, пов'язаних з роботою.

Загальний огляд здоров'я: крім специфічних оглядів, працівникам, які працюють за ПК, також необхідно регулярно проходити загальний медичний огляд, щоб виявити будь-які загрози для їхнього загального здоров'я.

Після проведення профілактичного медичного огляду для працівників ПК, медичний спеціаліст повинен надати підходящі рекомендації та попередження щодо збереження здоров'я працівника. Це можуть бути вимоги щодо режиму роботи за комп'ютером, правильного харчування, фізичних вправ, психологічного благополуччя та інших факторів, що сприяють збереженню здоров'я.

Конкретні вимоги можуть відрізнятися в залежності від країни, законодавства та специфіки робочого місця. Необхідно звернутися до медичної служби або органу з охорони праці в організації для отримання докладнішої інформації про вимоги до профілактичних медичних оглядів для працівників ПК.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК

У рамках даної кваліфікаційної роботи була розроблена комп'ютеризована система інформування про прийом медикаментів на основі Arduino UNO. Основною метою цього проекту було створення простої, ефективної та доступної системи, яка нагадуватиме користувачеві про потрібний режим прийому ліків та забезпечуватиме контроль за виконанням цього режиму.

В ході роботи була проведена розробка апаратної та програмної частини системи. Для апаратної реалізації було використано Arduino UNO, яка забезпечувала зручність підключення необхідних компонентів та була економічно доступною. Були використані модуль годинника RTC, рідкокристалічний дисплей та динамік для звукового сповіщення користувача.

Програмне забезпечення було розроблено на мові C++, в середовищі Arduino IDE, використовуючи бібліотеки LiquidCrystal.h, EEPROM, RTCLib.h та Wire.h. Були реалізовані функції для програмування розкладу прийому ліків, збереження цих даних у постійну пам'ять EEPROM та виведення повідомлень на дисплей та активатор звукових сповіщень у встановлені години.

Проведене тестування системи показало її ефективність та надійність. Користувач може зручно програмувати розклад прийому ліків та отримувати своєчасні нагадування. Система дозволяє забезпечити контроль за виконанням прийому ліків та зменшити ризик пропуску дози.

Отже, комп'ютеризована система інформування про прийом медикаментів на основі Arduino UNO виявилася успішною у виконанні своїх функцій. Вона може бути використана як простий інструмент для організації режиму прийому ліків та нагадування про нього, що робить її корисною для широкого кола користувачів.

Завершуючи, робота відображає значимість використання мікроконтролерних платформ, таких як Arduino UNO, для розробки простих, але ефективних медичних систем. Дана система може слугувати основою для подальших вдосконалень та розширень з метою створення більш складних та функціональних рішень у галузі медицини та догляду за здоров'ям.

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Webster J. G. Medical Instrumentation: Application and Design. Wiley & Sons, Incorporated, John, 2009.

2. Arduino офіційна документація – документація Arduino UNO та пов'язаних компонентів, яка надає детальну інформацію про платформу, її функції та використання. URL: <https://www.arduino.cc/reference/en/> (дата звернення: 17.04.2023)

3. Документація бібліотеки EEPROM – офіційна документація бібліотеки для роботи з постійною пам'яттю EEPROM в Arduino, яка дозволяє зберігати дані навіть після вимкнення живлення. URL: <https://www.arduino.cc/en/Reference/EEPROM> (дата звернення: 17.04.2023)

4. RTC DS3231 datasheet. URL: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf> (дата звернення: 17.04.2023)

5. LCD 1602 datasheet. URL: <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf> (дата звернення: 18.04.2023)

6. Документація бібліотеки LiquidCrystal.h – офіційна документація бібліотеки, що використовується для роботи з рідиннокристалічними дисплеями в Arduino. URL: <https://www.arduino.cc/en/Reference/LiquidCrystal> (дата звернення: 18.04.2023)

7. Документація бібліотеки Wire.h – офіційна документація бібліотеки для роботи з шиною I2C (TWI) в Arduino, яка використовується для зв'язку з реальним часом модулів (RTC) та інших пристроїв. URL: <https://www.arduino.cc/en/Reference/Wire> (дата звернення: 17.04.2023)

8. Документація бібліотеки RTCLib – документація бібліотеки, що надає функціональність для роботи з модулями реального часу (RTC) в Arduino. URL: <https://github.com/adafruit/RTCLib> (дата звернення: 17.04.2023)

					<i>КС КРБ 123.034.00.00 ПЗ</i>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Blum R. Arduino Programming in 24 Hours, Sams Teach Yourself. Pearson Education, Limited.

10. Margolis M. Arduino Cookbook. Sebastopol, California : O'Reilly Media, 2011.

11. Осухівська Г. М., Тиш Є. В., Луцик Н. С., Паламар А. М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

12. Проведення обов'язкових профілактичних медичних оглядів працівників окремих професій. URL: <https://zakon.rada.gov.ua/laws/show/z0639-02#Text>

13. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі. [навчальний посібник] Львів: «Магнолія 2006». 2013. 256с.

14. Купратий І.Г., Паламар А.М. Комп'ютерна система для дистанційного моніторингу стану здоров'я пацієнтів. Актуальні задачі сучасних технологій : збірник тез доповідей XI міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 7-8 грудня 2022 року), Тернопіль: ФОП Паляниця В. А., 2022. С. 142.

15. Паламар А.М., Купратий І.Г. Система для дистанційного моніторингу стану здоров'я пацієнтів на основі інтернету медичних речей. Матеріали X науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 7-8 грудня 2022 року), Тернопіль: ТНТУ, 2022. С. 85.

16. Скобло Ю.С., Соколовська Т.Б., Морозенко Д.І. та ін. Безпека життєдіяльності. Навчальний посібник для вищих навчальних закладів 3-4 рівнів акредитації. – К.: Кондор, 2003. 424 с.

17. Стищенко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. «Безпека життєдіяльності»: навч. посібник / Т.Є Стищенко, Г.В. Пронюк, Н.М. Сердюк, І.І. Хондак. – Харків: ХНУРЕ, 2018. – 336 с.

					КС КРБ 123.034.00.00 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«Затверджую»

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ___ ” _____ 2023р.

Комп'ютеризована система інформування про прийом ліків на основі Arduino UNO

ТЕХНІЧНЕ ЗАВДАННЯ

на ___ листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ Зав. Кафедри КС Осухівська Г.М.

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

_____ Бражніков В.А.

Тернопіль 2023

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система інформування про прийом медикаментів на основі Arduino UNO».

Умовне позначення кваліфікаційної роботи: КС КРБ 213.034.00.00

1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерної інженерії, Тернопільського національного технічного університету імені Івана Пулюя, Бражніков Владислав Андрійович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-238).

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання роботи – 28.02.2023 р.

Плановий термін завершення виконання роботи – 22.06.2023 р.

1.5 Порядок оформлення та представлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ЄСКД, ЄСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90%, наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначенні і цілі створення розробки

2.1 Призначення розробки

Комп'ютеризована система інформування про прийом медикаментів на основі Arduino UNO є дуже простою у користуванні. За допомогою кнопок на макетній платі користувач обирає кількість нагадувань на добу та отримує сповіщення у заданий програмним кодом час.

До складу системи повинні входити як апаратна складова, так і програмна.

2.2 Мета створення системи

Метою проєкту є розробка системи з інформуванням про прийом медикаментів.

2.3 Характеристика об'єкту

Система проєктується для інформування користувача, що включає в себе:

- Розробку структурної схеми;
- Розробку електричної принципової схеми
- Розробку блок-схеми алгоритму та програмного забезпечення для мікроконтролера

3 Вимоги до системи

3.1 Загальні вимоги

Комп'ютеризована система інформування про прийом медикаментів повинна забезпечити:

- Компактність: Система не повинна займати багато простору
- Низька собівартість
- Можливість вибору декількох режимів оповіщення
- Нагадування про прийом медикаментів: Система повинна мати здатність нагадувати користувачу про необхідність прийняти медикаменти відповідно до заданого графіку.
- Звукові сигнали: Система повинна мати можливість видавати звукові сигнали для нагадування про прийом медикаментів.

3.1.1 Вимоги до структури та функціонування системи

Структура системи інформування про прийом медикаментів включає в себе:

- RTC DS3231 – модуль визначення реального часу, який отримує та обробляє дані про час та надає програмі дані для видачі сповіщень
- 1602 LCD – дисплей для виводу тексту, що складається з 16 символних стовпчиків та 2 рядків та може відображати до 32 символів одночасно.
- Arduino UNO – основний компонент всього приладу, який займається зчитуванням, обробкою, виводом даних та сповіщень.

3.1.2 Вимоги в функціонуванні системи

Система функціонує у режимі, який налаштовує користувач.

3.1.3 Вимоги до надійності системи

Система має бути захищена від фізичних або механічних пошкоджень, як на рівні апаратного, так і програмного забезпечення. Надійність системи повинна

забезпечувати можливість відновлення її функціонування у випадку виникнення збоїв на рівні апаратних компонентів або програмного забезпечення. Ймовірність безвідмовної роботи системи повинна становити не менше 95%.

3.1.4 Вимоги до функцій та задач, які виконує система

Основна функціональна задача системи полягає у відправленні сповіщень користувачу в заданий інтервал часу.

3.1.5 Вимоги до апаратного забезпечення

- Режим роботи і умови експлуатації вибраних елементів повинні відповідати вказаним в ТЗ
- Елементна база має забезпечувати необхідні режими роботи системи

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації складається з:

- Пояснювальної записки
- Графічного матеріалу:
 - Структурної схеми
 - Електрично-принципової схеми
 - Блок-схеми
 - Програмного інтерфейсу

*Примітка: у комплект документації можуть вноситися зміни та доповнення в процесі розробки.

5 Техніко-економічні показники

Планована собівартість проєкту повинна становити не більше 3 тисяч гривень.

*Примітка: собівартість може змінюватися під час розрахунку в процесі розробки.

6 Стадії та етапи проєктування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	01.03 – 08.03.2023
2	Аналіз технічного завдання	10.03 – 24.03.2023
3	Аналіз та обґрунтування можливих рішень	28.03 – 03.04.2023
4	Розробка структурної схеми	04.04 – 10.04.2023
5	Розробка електрично-принципової схеми, вибір елементної бази	
6	Розробка програмного забезпечення для проєктованої системи	11.04 – 21.04.2023
7	Безпека життєдіяльності, основи охорони праці	22.04 – 28.04.2023
8	Оформлення кваліфікаційної роботи	01.06 – 08.06.2023
9	Попередній захист кваліфікаційної роботи	16.06 – 18.06.2023
10	Захист кваліфікаційної роботи	23.06.2023

7 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Перелік елементів

Поз. Позначення	Найменування	Кіл.	Примітка
	Макетна плата	1	
	Динаміки		
BA1	Active Buzzer	1	
	Кнопки		
K1..K4	ПКН-2	4	
	Світлодіоди		
LED1	HL43-2	5	
	Резистори		
R1..R4	10кОм	4	
R4,R21	1кОм	17	
	Мікросхеми		
U4	Arduino UNO	1	
U5	RTC DS3231	1	
U3	ATMEGA8U2-MU	1	
ZIC1	ATMEGA8	1	
	Дисплеї		
LCD1	LCD 1602	1	
	Конденсатори		
C1..C14	1u – 100n	14	
	Запобіжники		
F1	500мА	1	
	Кварцові резонатори		
Q1, Q2	16МГц	2	
	Варистор		
Z1, Z2	12В	2	

КС КРБ 123.034.00.00 ПЕ

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Бражніков В.А.			Комп'ютеризована система інформування про прийом медикаментів Перелік елементів	Літ.	Арк.	Акрушіє
Перевір.		Осухівська Г.М.					60	
Реценз.		Млинко Б.Б.				ТНТУ, каф. КС, гр.СІ-41		
Н. контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

Додаток В

Лістинг програми «Комп'ютеризована система інформування про прийом медикаментів»

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <RTClib.h>
#include <EEPROM.h>

int pushVal = 0;
int val;
int val2;
int addr = 0;

RTC_DS3231 rtc;

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
// lcd pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define getWellsoon 0
#define HELP_SCREEN 1
#define TIME_SCREEN 2

//bool pushPressed;           //flag to keep track of push button state
int pushpressed = 0;

const int ledPin = LED_BUILTIN;           // buzzer and led pin
int ledState = LOW;
int Signal = 0;
```

```

int buzz = 13;

int push1state, push2state, push3state, stopinState = 0;

int push1Flag, push2Flag, Push3Flag = false;    // push button flags

int push1pin = 9;

int push2pin = 8;

int push3pin = 7;

int stopPin = A0;

int screens = 0;           // screen to show

int maxScreen = 2;        // screen count

bool isScreenChanged = true;

long previousMillis = 0;

long interval = 1000;           // buzzing interval

unsigned long currentMillis;

long previousMillisLCD = 0;    // for LCD screen update

long intervalLCD = 5000;       // Screen cycling interval

unsigned long currentMillisLCD;

// Set Reminder Change Time

int buzz8amHH = 8;           // HH - hours           ##Set these for
reminder time in 24hr Format

int buzz8amMM = 00;          // MM - Minute

int buzz8amSS = 00;          // SS - Seconds

int buzz2pmHH = 8;           // HH - hours

int buzz2pmMM = 01;          // MM - Minute

int buzz2pmSS = 00;          // SS - Seconds

```

```

int buzz8pmHH = 8;           // HH - hours
int buzz8pmMM = 02;         // MM - Minute
int buzz8pmSS = 00;         // SS - Seconds

int stop8am = 0;
int stop2pm=0;
int stop8pm=0;

int nowHr, nowMin, nowSec;   // to show current
mm,hh,ss

// All messeges

void gwsMessege() {         // print get well soon messege

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Stay Healthy :)"); // Give some cheers

    lcd.setCursor(0, 1);

    lcd.print("Get Well Soon :)"); // wish

}

void helpScreen() {         // function to display 1st screen in
LCD

    lcd.begin(16,2);

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Press Buttons");

    lcd.setCursor(0, 1);

    lcd.print("for Reminder...!");

}

void timeScreen() {         // function to display Date and time
in LCD screen

```



```

    DateTime now = rtc.now();           // take rtc time and print in
display
    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Time:");

    lcd.setCursor(6, 0);

    lcd.print(nowHr = now.hour(), DEC);

    lcd.print(":");

    lcd.print(nowMin = now.minute(), DEC);

    lcd.print(":");

    lcd.print(nowSec = now.second(), DEC);

    lcd.setCursor(0, 1);

    lcd.print("Date: ");

    lcd.print(now.day(), DEC);

    lcd.print("/");

    lcd.print(now.month(), DEC);

    lcd.print("/");

    lcd.print(now.year(), DEC);
}

void setup() {

    Serial.begin(9600);                 // start serial debugging

    if (! rtc.begin()) {                // check if rtc is
connected

        Serial.println("Couldn't find RTC");

        while (1);

    }

    if (rtc.lostPower()) {

```

```

    Serial.println("RTC lost power, lets set the time!");
}

// rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));           //
// uncomment this to set the current time and then comment in next upload
// when u set the time

rtc.adjust(DateTime(2019, 1, 10, 7, 59, 40));                //
manual time set

lcd.begin(16, 2);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Welcome To");                                   //
print a messege at startup

lcd.setCursor(0, 1);

lcd.print("Circuit Digest");

delay(1000);

pinMode(push1pin, INPUT);                                  //
define push button pins type

pinMode(push2pin, INPUT);

pinMode(push3pin, INPUT);

pinMode(stopPin, INPUT);

pinMode(ledPin, OUTPUT);

delay(200);

Serial.println(EEPROM.read(addr));

val2 = EEPROM.read(addr);                                  // read previously
saved value of push button to start from where it was left previously

switch (val2) {

    case 1:

        Serial.println("Set for 1/day");

        push1state = 1;

```

```
    push2state = 0;

    push3state = 0;

    pushVal = 1;

    break;
case 2:

    Serial.println("Set for 2/day");

    push1state = 0;

    push2state = 1;

    push3state = 0;

    pushVal = 2;

    break;
case 3:

    Serial.println("Set for 3/day");

    push1state = 0;

    push2state = 0;

    push3state = 1;

    pushVal = 3;

    break;
}

}

void loop() {

    push1(); //call to set
    once/day

    push2(); //call to set
    twice/day
```

```

    push3(); //call to set
    thrice/day

    if (pushVal == 1) { // if push
    button 1 pressed then remind at 8am

        at8am(); //function to
        start buzzing at 8am

    }

    else if (pushVal == 2) { // if push
    button 2 pressed then remind at 8am and 8pm

        at8am();

        at8pm(); //function to
        start uzzing at 8mm

    }

    else if (pushVal == 3) { // if push
    button 3 pressed then remind at 8am and 8pm

        at8am();

        at2pm(); //function to
        start uzzing at 8mm

        at8pm();

    }

    currentMillisLCD = millis(); // start millis
    for LCD screen switching at defined interval of time

    push1state = digitalRead(push1pin); // start
    reading all push button pins

    push2state = digitalRead(push2pin);

    push3state = digitalRead(push3pin);

    stopinState = digitalRead(stopPin);

    stopPins(); // call to
    stop buzzing

    changeScreen(); // screen
    cycle function

```

```

}

// push buttons

void push1() { // function to set reminder once/day

    if (push1state == 1) {

        push1state = 0;

        push2state = 0;

        push3state = 0;

        // pushpressed = true;

        EEPROM.write(addr, 1);

        Serial.print("Push1 Written : ");
        Serial.println(EEPROM.read(addr)); // for debugging

        pushVal = 1; //save
the state of push button-1

        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("Reminder set ");

        lcd.setCursor(0, 1);

        lcd.print("for Once/day !");

        delay(1200);

        lcd.clear();

    }

}

void push2() { //function to set reminder
twice/day

    if (push2state == 1) {

        push2state = 0;

```

```

    push1state = 0;

    push3state = 0;

    // pushpressed = true;

    digitalWrite(addr, 2);

    Serial.print("Push2 Written : ");
    Serial.println(digitalRead(addr));

    pushVal = 2;

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Reminder set ");

    lcd.setCursor(0, 1);

    lcd.print("for Twice/day !");

    delay(1200);

    lcd.clear();

}

}

void push3() { //function to set reminder
    thrice/day

    if (push3state == 1) {

        push3state = 0;

        push1state = 0;

        push2state = 0;

        // pushpressed = true;

        EEPROM.write(addr, 3);

        Serial.print("Push3 Written : ");
        Serial.println(EEPROM.read(addr));

        pushVal = 3;

        lcd.clear();

```

```

    lcd.setCursor(0, 0);

    lcd.print("Reminder set ");

    lcd.setCursor(0, 1);

    lcd.print("for Thrice/day !");

    delay(1200);

    lcd.clear();

}

}

void stopPins() { //function to stop buzzing when
user pushes stop push button

    if (stopinState == 1) {
//    stopinState = 0;
//    pushPressed = true;
    pushpressed = 1;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Take Medicine ");
    lcd.setCursor(0, 1);
    lcd.print("with Warm Water");
    delay(1200);
    lcd.clear();
    DateTime t = rtc.now();
if(int(t.hour()) == buzz8amHH && int(t.minute()) == buzz8amMM) {
    stop8am=1;
    ledState = LOW;
    digitalWrite(ledPin, ledState);
}
if(int(t.hour()) == buzz2pmHH && int(t.minute()) == buzz2pmMM) {
    stop2pm=1;
    ledState = LOW;
    digitalWrite(ledPin, ledState);
}
if(int(t.hour()) == buzz8pmHH && int(t.minute()) == buzz8pmMM) {
    stop8pm=1;
    ledState = LOW;
    digitalWrite(ledPin, ledState);
}

}

}

void startBuzz() { // function to start buzzing
when time reaches to defined interval

```

```

// if (pushPressed == false) {
// if (pushpressed == 0) {
    Serial.println("pushpressed is false in blink");
    unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;          // save the last time
you blinked the LED
    Serial.println("Start Buzzing");

if (ledState == LOW) {                          // if the LED is off turn it
on and vice-versa:
    ledState = HIGH;
    } else {
    ledState = LOW;
    }
    digitalWrite(ledPin, ledState);
    }
}
void at8am() {                                  // function to start buzzing at
8am

    DateTime t = rtc.now();

    if (int(t.hour()) == buzz8amHH && int(t.minute()) == buzz8amMM &&
(int(t.second()) == buzz8amSS || int(t.second()) <
buzz8amSS+10)&&stop8am!=1) {

        // if (int(now.hour()) >= buzz8amHH) {

        //   if (int(now.minute()) >= buzz8amMM) {

            // if (int(now.second()) >= buzz8amSS) {
                // if(stopinState!=1){

startBuzz();
                // }

                //////////////////////////////////////

                // buzz8amSS= buzz8amSS+30;

                //////////////////////////////////////

            // }

        // }

    }
}
}

```



```

        // }

//    }

// }

}

//Screen Cycling

void changeScreen() {                                //function for Screen Cycling

    // Start switching screen every defined intervalLCD

    if (currentMillisLCD - previousMillisLCD > intervalLCD)
// save the last time you changed the display

    {

        previousMillisLCD = currentMillisLCD;

        screens++;

        if (screens > maxScreen) {

            screens = 0; // all screens over -> start from 1st

        }

        isScreenChanged = true;

    }

    // Start displaying current screen

    if (isScreenChanged) // only update the screen if the screen is
changed.

    {

        isScreenChanged = false; // reset for next iteration

        switch (screens)

        {

            case getWellsoon:

                gwsMessege(); // get well soon message

                break;

```

```
case HELP_SCREEN:  
    helpScreen();           // instruction screen  
    break;  
  
case TIME_SCREEN:  
    timeScreen();          // to print date and time  
    break;  
  
default:  
    timeScreen();  
    break;  
  
}  
  
}  
  
}
```