

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет Інформаційних електронних систем

(повне найменування інституту, назва факультету (відділення))

Кафедра Інформаційних радіоелектронних технологій і систем

(повна назва кафедри (предметної, циклової комісії))

## БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему: «Радіотехнічна система відображення інформації на сервоприводах»

Виконав: студент 2(4)-го курсу, групи ТКР-21мс  
спеціальності 172 Телекомунікації та  
радіотехніка

(шифр і назва напрямку підготовки, спеціальності)

Пехота Р.В.

(прізвище та ініціали)

Керівник: к.т.н., ст.викл. каф. ІРТС

Притула М.О.

(прізвище та ініціали)

«15» 06 2023 р.

Рецензент: д.т.н., професор каф. ІКСТ

Михалевський Д.В.

(прізвище та ініціали)

«16» 06 2023 р.

**Допущено до захисту**

**Завідувач кафедри ІРТС**

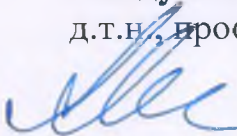
**д.т.н./проф. Осадчук О.В.**

(прізвище та ініціали)

«16» 06 2023 р.

Вінницький національний технічний університет  
 Факультет Інформаційних електронних систем  
 Кафедра Інформаційних радіоелектронних технологій і систем  
 Рівень вищої освіти перший (бакалаврський)  
 Галузь знань – 17 Електроніка та телекомунікації  
 Спеціальність – 172 – Телекомунікації та радіотехніка  
 Освітньо-професійна програма – Радіотехніка

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри ІРТС**  
 д.т.н., проф. Осадчук О.В.

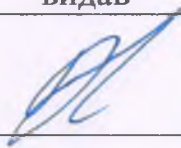
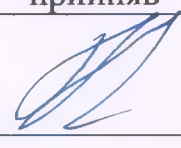


 21.03.2023 року

## ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Пехоті Роману Віталійовичу  
 (прізвище, ім'я, по батькові)

1. Тема роботи. Радіотехнічна система відображення інформації на сервоприводах  
 керівник роботи к.т.н., ст.викл. каф. ІРТС Притула М.О.  
 затверджені наказом вищого навчального закладу від "20"03 2023 року № 67
2. Строк подання студентом роботи 16 06 2023 року
3. Вихідні дані до роботи: Тип індикації – блінкерна, можливість відображення різного типу інформації, встановлення поточного часу – через персональний комп'ютер, наявність модуля РТС, збереження поточного часу при відключенні живлення, живлення –зовнішній адаптер.
4. Зміст текстової частини: Вступ. Аналіз радіотехнічної системи відображення інформації на сервоприводах. Розробка модулів радіотехнічної системи відображення інформації на сервоприводах. Моделювання радіотехнічної системи відображення інформації на сервоприводах. Охорона праці. Висновки. Список використаних джерел
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):  
Структурна схема Радіотехнічної системи відображення інформації на сервоприводах. Функціональна схема Радіотехнічної системи відображення інформації на сервоприводах. Схема індикації Радіотехнічної системи відображення інформації. Електронний макет для експериментальних досліджень. Схема електрична принципова та перелік елементів.

## 6. Консультанти розділів роботи

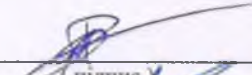
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	к.т.н., ст.викл. каф. ІРТС Притула М.О.		
Охорона праці	професор кафедри БЖДПБ, професор д.п.н., Дембіцька С.В.		

7. Дата видачі завдання 22.03. 2023 року

## КАЛЕНДАРНИЙ ПЛАН


№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Вибір, узгодження та затвердження теми БДР	14.02.2023-28.02.2023	
2.	Огляд та аналіз літературних джерел.	01.03.2023-23.03.2023	
3.	Затвердження теми. Розробка завдання на БДР.	20.03.2023-31.03.2023	
4.	Попередня розробка основних розділів. Аналіз вирішення поставленої задачі. Розробка структурної схеми та технічних рішень.	01.04.2023-06.05.2023	
5.	Математичне моделювання та електричні розрахунки. Експериментальне дослідження.	07.05.2023-18.05.2023	
6.	Розробка ілюстративної частини БДР	19.05.2023-22.05.2023	
7.	Охорона праці (ОП)	23.05.2023-28.05.2023	
8.	Оформлення пояснювальної записки та ілюстративної частини.	29.05.2023-06.06.2023	
9.	Нормоконтроль	07.06.2023-09.06.2023	
10.	Попередній захист БДР, доопрацювання, рецензування БДР	10.06.2023-19.06.2023	
11.	Захист БДР ЕК	20.06.2023-21.06.2023	

Студент

  
 (підпис)

Пехота Р.В.

Керівник роботи

  
 (підпис)

Притула М.О.

## АНОТАЦІЯ

УДК 621.371

Пехота Р.В. Радіотехнічна система відображення інформації на сервоприводах. Бакалаврська дипломна робота зі спеціальності 172 - Телекомунікації та радіотехніка, освітня програма – Радіотехніка. Вінниця: ВНТУ, 2023. 82 с. Укр. мовою. Бібліогр.: 26 назв; рис.: 35; табл. 6.

Бакалаврська дипломна робота присвячена розробці радіотехнічної системи відображення інформації на сервоприводах. Основною метою дослідження є розробка та впровадження системи, яка дозволить точно відображати різні типи даних на рухомих частинах з використанням семисегментної індикації. Робота базується на модулі Arduino Nano та взаємодії з сервоприводами за допомогою шини I2C.

У першому розділі проведено аналіз радіотехнічних систем відображення інформації на сервоприводах. Розглянуті алгоритми керування та особливості взаємодії модуля Arduino Nano з сервоприводами через шину I2C. Результати аналізу підтвердили актуальність та перспективи використання даної системи.

В роботі виконано розробку та реалізацію радіотехнічної системи відображення інформації на сервоприводах. Розроблені модулі системи взаємодіють між собою за допомогою шини I2C, алгоритми керування забезпечують точне відображення різних типів даних. Результати експериментів підтверджують працездатність та ефективність розробленої системи.

Застосування радіотехнічної системи відображення інформації на сервоприводах може бути широким, зокрема в промисловості, автомобільній техніці, побутових пристроях та інших галузях, де точна візуалізація даних є важливою.

Ключові слова: радіотехнічна система, серводвигуни, модулі, обробка інформації.

## ABSTRACT

Pekhota R.V. Mobile Radio-Technical System for Environmental Parameter Monitoring. Bachelor's thesis in the specialty 172 - Telecommunications and Radio Engineering, educational program - Radio Engineering. Vinnytsia: VNTU, 2023. 82 p.

Language: Ukrainian. Bibliography: 26 titles; figures: 35; tables: 6.

The bachelor's diploma thesis is dedicated to the development of a radio-technical system for information display on servo drives. The main objective of the research is to design and implement a system that allows accurate display of various types of processing. data on moving parts using seven-segment indication. The system is based on the Arduino Nano module and utilizes the I2C bus for communication with the servo drives.

The first section of the thesis provides an analysis of radio-technical systems for information display on servo drives. It covers control algorithms and the peculiarities of interaction between the Arduino Nano module and servo drives through the I2C bus. The analysis confirms the relevance and potential of using this system.

The thesis focuses on the development and implementation of the radio-technical system for information display on servo drives. The designed system modules interact with each other through the I2C bus, and the control algorithms ensure precise display of various data types. The experimental results demonstrate the functionality and efficiency of the developed system.

The application of the radio-technical system for information display on servo drives has broad potential in industries such as manufacturing, automotive technology, household devices, and other areas where accurate data visualization is crucial.

Overall, this diploma thesis presents the design, implementation, and analysis of a radio-technical system that enables accurate information display on servo drives. The findings highlight the practicality and significance of such a system in various domains, showcasing its potential for future applications and advancements.

Keywords: radio-technical system, servomotors, modules, information

## ЗМІСТ

<b>ВСТУП .....</b>	<b>7</b>
<b>1 АНАЛІЗ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ .....</b>	<b>9</b>
1.1 Аналіз конструкційних рішень .....	9
1.2 Актуальність та переваги радіотехнічної системи відображення інформації на сервоприводах .....	10
1.3 Висновки до розділу .....	11
<b>2 РОЗРОБКА МОДУЛІВ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ .....</b>	<b>13</b>
2.1 Вибір та обґрунтування схеми системи відображення інформації на сервоприводах.....	13
2.2 Вибір елементів індикації та схеми керування системи відображення інформації на сервоприводах .....	21
2.3 Розробка програмного забезпечення.....	24
<b>3 МОДЕЛЮВАННЯ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ .....</b>	<b>35</b>
3.1 Вибір моделюючого пакету.....	35
3.2 Моделювання рухомого блоку радіотехнічної системи відображення інформації на сервоприводах .....	38
3.3 Дослідження сигналів комунікації модулів I2C.....	40
<b>4 ОХОРОНА ПРАЦІ .....</b>	<b>46</b>
4.1 Технічні рішення з безпечного виконання роботи .....	47
4.2 Технічні рішення з гігієни праці та виробничої санітарії .....	52
4.3 Пожежна безпека.....	58
<b>ВИСНОВКИ .....</b>	<b>61</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>63</b>
Додаток А (обов'язковий). Протокол перевірки навчальної (бакалаврської) роботи .....	65
Додаток Б (обов'язковий). Ілюстративний матеріал .....	67
Додаток В (довідниковий). Лістинг програми для Радіотехнічної системи відображення інформації на сервоприводах.....	74

## ВСТУП

У сучасному світі, де швидкість обміну інформацією є вирішальним чинником, виникає все більша потреба в розробці ефективних та надійних радіотехнічних систем, спроможних швидко та точно відображати інформацію. Одним із найважливіших напрямків в цьому контексті є розробка радіотехнічних систем відображення інформації на сервоприводах. Ця технологія дозволяє реалізувати взаємодію між користувачем і системою шляхом передачі та відображення даних на рухомих механізмах.

### *Актуальність теми.*

В сучасному світі, коли швидкість та точність передачі інформації мають вирішальне значення у різних галузях, таких як промисловість, автоматизація, медицина, телекомунікації та інші, радіотехнічні системи відображення інформації на сервоприводах набувають все більшої актуальності. Ці системи забезпечують ефективний спосіб передачі та відображення інформації [1], що відображається на рухомих механізмах, таких як сервоприводи. Вони дозволяють створювати інтерактивні інтерфейси, де користувач може спостерігати рухи механізму, що точно відповідають переданій інформації. Це дозволяє зробити процес комунікації більш інтуїтивно зрозумілим і забезпечує взаємодію на новому рівні.

*Метою роботи* є розробка та вивчення радіотехнічної системи відображення інформації на сервоприводах з метою розширення їх можливостей та оптимізації процесів комунікації.

Типи інформації та їх відображення: Радіотехнічна система відображення інформації на сервоприводах забезпечує передачу різних типів інформації таку як: текстову та числову. Текстова інформація може бути відображена у вигляді рухомих літер, слів або повідомлень, що надходять в режимі реального часу. Числова інформація може бути відображена у вигляді числових значень, показників або діаграм, що допомагає зрозуміти числові дані та тренди.

*Об'єктом дослідження* є радіотехнічна система відображення інформації на сервоприводах, зокрема її апаратна та програмна складові. Апаратна складова

об'єкта включає самі сервоприводи, які виконують функцію відображення, а також засоби зв'язку, необхідні для передачі сигналів і керування системою[2]. Програмна складова об'єкта включає розроблене програмне забезпечення, що забезпечує керування сервоприводами, генерацію сигналів та відображення необхідної інформації.

*Предметом дослідження* є можливості, типи і переваги відображення інформації на сервоприводах у радіотехнічній системі, зокрема можливість відображення дати та часу за допомогою семисегментної конструкції індикації.

*Для досягнення поставленої мети необхідно вирішити наступні задачі:*

1. Аналіз і вивчення науково-технічної літератури та джерел, що стосуються радіотехнічних систем відображення інформації на сервоприводах.
2. Розробка архітектури та алгоритмів радіотехнічної системи відображення інформації на сервоприводах.
3. Розробка та реалізація необхідного програмного забезпечення для управління радіотехнічною системою.
4. Проведення експериментальних досліджень та оцінка ефективності розробленої системи.
5. Аналіз результатів експериментів та висновки щодо досягнутих результатів та можливостей подальшого вдосконалення радіотехнічної системи відображення інформації на сервоприводах.



# 1 АНАЛІЗ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ

## 1.1 Аналіз конструкційних рішень

Огляд семисегментної індикації:

У цьому підпункті проведемо аналіз семисегментної індикації як основного методу відображення інформації на сервоприводах. Розглянемо принцип роботи семисегментної індикації, її структуру [3] та можливості відображення різних символів та чисел. Зазначимо переваги цього методу, такі як висока читабельність, простота конструкції та широке застосування в різних пристроях відображення інформації.

Застосування рухомих частин в радіотехнічній системі:

В даному підпункті розглянемо використання рухомих частин, зокрема сервоприводів, як складової частини радіотехнічної системи відображення [4] інформації. Проаналізуємо принцип роботи сервоприводів, їх можливості та обмеження. Вказатимемо на переваги використання сервоприводів для відображення інформації, зокрема точність позиціонування, можливість керування кутом обертання та швидкістю руху.

Використання шини I2C для взаємодії модулів системи

У цьому підпункті розглянемо шину I2C (Inter-Integrated Circuit) [5] як засіб взаємодії між різними модулями радіотехнічної системи. Проаналізуємо принцип роботи шини I2C, її переваги, такі як простота підключення та можливість підключення багатьох пристроїв до одного інтерфейсу. Вказатимемо на сумісність шини I2C з модулем Arduino Nano

Роль модуля Arduino Nano в радіотехнічній системі

У даному підпункті розглянемо роль модуля Arduino Nano [6] у радіотехнічній системі відображення інформації на сервоприводах. Проаналізуємо можливості та функціональні можливості Arduino Nano, які дозволяють йому виконувати роль керуючого пристрою для системи відображення. Вказатимемо на його здатність підключати та керувати різними пристроями через шину I2C, що

робить його ідеальним вибором для взаємодії з семисегментною індикацією та сервоприводами.

## 1.2 Актуальність та переваги радіотехнічної системи відображення інформації на сервоприводах

В цьому підпункті підкреслимо актуальність і важливість розробки радіотехнічної системи відображення інформації на сервоприводах. Зазначимо, що така система може бути застосована в різних сферах, включаючи автоматизацію, візуалізацію даних та дисплеї у промислових, транспортних та побутових пристроях. Вказатимемо на переваги такої системи, такі як точне та зручне відображення інформації, можливість відображення різних типів даних (числа, символи, дата та час), гнучкість у налаштуванні та можливість інтеграції з іншими пристроями.

### Постановка задач дослідження

У цьому підпункті сформулюємо конкретні задачі, які необхідно вирішити для досягнення мети роботи. Зазначимо наступні задачі:

Аналіз семисегментної індикації та визначення способів відображення різних типів інформації (числа, символи, дата та час) на сервоприводах.

Вивчення принципу роботи та параметрів сервоприводів для оптимального використання їх у системі відображення інформації.

Розробка алгоритмів керування сервоприводами для точного позиціонування та керування кутом обертання.

Дослідження можливостей шини I2C для взаємодії модулів системи та розробка відповідного програмного забезпечення для керування системою через цю шину.

Встановлення взаємозв'язку між модулем Arduino Nano, сервоприводами та семисегментною індикацією для успішної реалізації системи відображення інформації.

Оптимізація системи відображення, включаючи визначення оптимальних налаштувань, швидкості руху сервоприводів та підбір оптимального розміщення рухомих частин системи.

Випробування та оцінка ефективності розробленої радіотехнічної системи відображення інформації на сервоприводах на різних типах даних та у різних умовах експлуатації.

Загальна мета цього розділу - провести аналіз радіотехнічної системи відображення інформації на сервоприводах, включаючи семисегментну індикацію, використання рухомих частин, шини I2C та модуль Arduino Nano. Також необхідно сформулювати задачі, які допоможуть вирішити поставлену мету дослідження та розробки.

### 1.3 Висновки до розділу

У даному розділі був проведений аналіз радіотехнічної системи відображення інформації на сервоприводах, яка базується на семисегментній індикації, використанні рухомих частин, шині I2C та модулі Arduino Nano. Оглянувши кожен з цих складових частин системи [6], було виявлено наступне:

Семисегментна індикація є ефективним методом відображення інформації, забезпечуючи високу читабельність та можливість відображення різних типів даних.

Використання сервоприводів дозволяє досягти точного позиціонування та керування кутом обертання рухомих частин системи, забезпечуючи гнучкість та широкі можливості відображення.

Шина I2C виявляється зручним засобом взаємодії між модулями системи, забезпечуючи простоту підключення та сумісність з модулем Arduino Nano.

Модуль Arduino Nano виконує важливу роль керуючого пристрою, забезпечуючи взаємозв'язок між семисегментною індикацією, сервоприводами та іншими модулями через шину I2C.

Актуальність системи відображення інформації на сервоприводах полягає в її потенціалі застосування у різних сферах, включаючи автоматизацію,

візуалізацію даних та вбудовані системи. Переваги цієї системи включають точне та зручне відображення інформації, гнучкість налаштування та інтеграцію з іншими пристроями, а також можливість відображення різних типів даних, включаючи числа, символи, дату та час. Ця система має потенціал для використання в промисловості, транспортних засобах, побутових пристроях та інших сферах, де точна та зручна відображення інформації є важливим фактором.

Для досягнення поставленої мети роботи, було сформульовано ряд задач, включаючи аналіз семисегментної індикації та визначення способів відображення різних типів інформації, вивчення принципу роботи та параметрів сервоприводів, розробку алгоритмів керування, дослідження можливостей шини I2C, встановлення взаємозв'язку між модулем Arduino Nano, сервоприводами та семисегментною індикацією, оптимізацію системи відображення та випробування розробленої системи.

У наступних розділах роботи будуть детальніше розглянуті ці аспекти, а також буде розроблено та впроваджено радіотехнічну систему відображення інформації на сервоприводах з використанням семисегментної індикації та модуля Arduino Nano. Результати цього дослідження і розробки можуть мати практичне застосування в різних галузях, де точна та зручна відображення інформації відіграє важливу роль.

## 2 РОЗРОБКА МОДУЛІВ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ

2.1 Вибір та обґрунтування схеми системи відображення інформації на сервоприводах

Радіотехнічна система відображення інформації на сервоприводах, що розробляється [7], повинна інформувати власника за допомогою чотирьох чисел, що формуються на чотирьох семисегментних модулях які відображають потрібну цифру підняттям та опусканням сегментів сервоприводами про час та дату, а також мати можливе збереження можливості запам'ятовувати інформацію для відображення при відсутності напруги живлення.

Для синтезу структурної схеми даного пристрою використовуються всі пункти технічного завдання:

Так як розробляється годинник, то потрібно забезпечити:

- 1) відображення годинника та його точний хід;
- 2) напруга живлення годинника 5В;
- 3) напруга живлення сервоприводів  $\approx 5\text{В}$ ;
- 4) можливість налаштування часу.

Блок, який обов'язково повинен бути в складі радіотехнічної системи відображення інформації на сервоприводах, - це пристрій індикації за допомогою сервоприводів. Для забезпечення можливості керування всіма сервоприводами використовуються драйвери керування. Для коректного відображення інформації використовується два драйвера керування сервоприводами які беруть на себе керування по дві «цифри» кожний, тобто години та хвилини. Загальна кількість сервоприводів В результаті радіотехнічна система відображення інформації на сервоприводах набуває наступного вигляду, структурна схема зображена в Додатку Б та на рисунку 2.1.

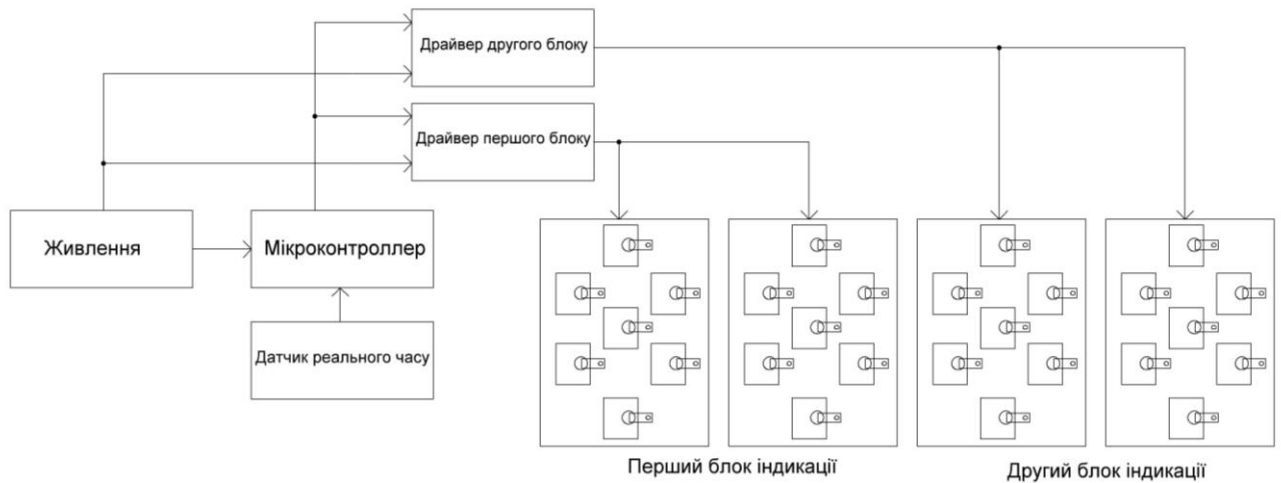


Рисунок 2.1 – Структурна схема радіотехнічної системи відображення інформації на сервоприводах

Вибір мікроконтролера та огляд його архітектури:

Щоб реалізувати схему радіотехнічної системи відображення інформації на сервоприводах існує немала кількість мікроконтролерів з різною архітектурою. Але для реалізації проекту буде вибиратись мікроконтролер з лінійки Arduino, через відносну простоту та особисті вподобання та зручність програмного середовища Arduino IDE. В лінійці Arduino існують такі моделі:

- 1) Arduino Mini;
- 2) Arduino Nano;
- 3) Arduino UNO;
- 4) Arduino Mega.

Досить доцільно використовувати мікроконтролер, який широко застосовується в цифровій техніці. Цей метод дозволяє отримати високу точність не прикладаючи надмірно високі вимоги до стабільності компонентів.

Для реалізації проекту був обраний готовий модуль-платформа Arduino Nano, який зображений на рисунку 2.2. Arduino Nano [6] є аналогом Arduino Uno, використовуючи той самий чіп ATmega328P, але має відмінний формфактор плати, що на 2-2,5 рази менший за Uno (53 x 69 мм). Завдяки своїм компактним розмірам, Arduino Nano дозволяє легко збирати складні схеми методом навісного

монтажу. Після стадії створення макету, ми збираємо функціонуючі примірники, і для цього Arduino Nano є найкращим варіантом. характеристики Arduino Nano:

- 1) мікроконтролер Atmel ATmega168 або ATmega328;
- 2) вхідна напруга (рекомендована) 7-12 В;
- 3) робоча напруга (логічна рівень) 5 В;
- 4) цифрові Входи / Виходи - 14 (6 з яких можуть використовуватися як виходи ШІМ);
- 5) аналогові входи – 8;
- 6) постійний струм через вхід / вихід 40 mA з одного виводу і 500 mA з усіх висновків;
- 7) флеш-пам'ять 16 Кб (ATmega168) або 32 Кб (ATmega328) при цьому 2 Кб використовуються для завантажувача;
- 8) EEPROM 512 байт (ATmega168) або 1 Кб (ATmega328);
- 9) ОЗП 1 Кб (ATmega168) або 2 Кб (ATmega328);
- 10) тактова частота 16 МГц;
- 11) розміри 1.85 см x 4.2 см.

Arduino Nano пропонує широкі можливості завдяки своїм характеристикам. Зокрема, він має 8 аналогових входів, які можуть також використовуватися як цифрові виходи. Крім цього, є 14 цифрових входів/виходів, з яких 6 можуть функціонувати як широтно-імпульсні модулятори (ШІМ). Додатково, є 2 входи підтримувані I2C та 3 підтримувані SPI. На протилежному кінці плати від роз'єму мікро-USB знаходиться колодка Arduino ICSP, яка призначена для прошивки мікроконтролера. Це зручна можливість для оновлення програмного забезпечення пристрою.

Завдяки вихідним ШІМ сигналам та наявності транзисторів, Arduino Nano може використовуватися для регулювання обертів двигуна, налаштування яскравості світлодіодів та виконання багатьох інших завдань. Аналогові входи дозволяють отримувати дані з аналогових датчиків, що розширює можливості пристрою.

Крім того, Arduino Nano має виходи Digital 2 і 3, які можуть використовуватися для зовнішніх переривань. Ці сигнали можуть сповістити

мікроконтролер про важливі події, викликаючи відповідну програму обробки переривань. Це дозволяє, наприклад, виходити з режиму енергозбереження та виконувати необхідні обчислення.

Arduino Nano може житися від різних джерел живлення, що робить його більш гнучким у використанні із різними конфігураціями та додатковими пристроями. У Arduino Nano [6] розташування виводів виконано так, як зображено на рисунку 2.2.

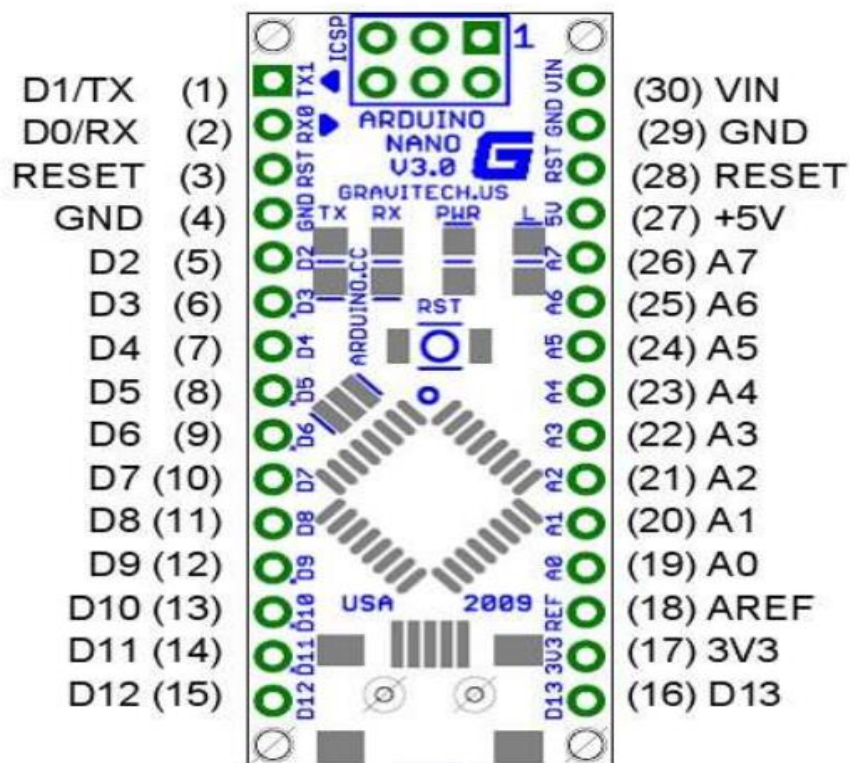


Рисунок 2.2 – Розташування виводів Arduino Nano

Призначення виводів:

- 1) 1 - TX (передача UART) або порт D0;
- 2) 2 - RX (прийом UART) або порт D1;
- 3) 3, 28 - скидання (RESET);
- 4) 4, 29 - земля;
- 5) 5 ... 16 - порти D3 ... D13;
- 6) 17 - напруга 3,3 В;



- 7) 18 - опорна напруга АЦП;
- 8) 19 ... 26 - 8 каналів АЦП А0 ... А7;
- 9) 27 - напруга 5,0 В;
- 10) 30 - плюс живлення модуля 2-20 В.

Перші два виводи використовуються або для зв'язку по класичному послідовному інтерфейсу з іншими пристроями, або як порти для двійкових даних. У Arduino Nano виводи 5...16, крім зазначених, мають додаткові функції:

- 1) 5 - переривання INT0;
- 2) 6 - переривання INT1 / ШІМ / АІН0;
- 3) 8 - таймер-лічильник Т1 / шина І2С SCL / ШІМ;
- 4) 7 - таймер-лічильник Т0 / шина І2С SDA / АІН1;
- 5) 9,12,13,14 - ШІМ;
- 6) 16 - світлодіод.

АІН0 і АІН1 - це входи швидкодіючого аналогового компаратора. Крім того, є 6 каналів з виходом широтно-імпульсного модулятора (ШІМ). До того ж є більша кількість контактів, на які можуть бути переведені запити переривань.

Вибір модуля RealTime та особливості його застосування:

Годинник обов'язково повинен мати пристрій керування часом, зберігання його ходу. Для реалізації годинника в простому випадку генератор імпульсів еталонної тривалості повинен виробляти хвилині імпульси, але реалізувати стабільний генератор такої тривалості достатньо складно. Якщо використовувати в якості задаючої ланки резистор номіналом 1 МОм і конденсатора ємністю 30 мкФ, то ми реалізуємо період повтора імпульсів 1 хв. Але стабільність такого генератора буде невисокою, на опір резистора буде впливати опір друкованої плати, а вона в свою чергу буде залежати від вологості повітря. А також особливо нестабільним елементом є конденсатор. Тому використаємо спеціальний модуль реального часу на мікросхемі DS3231 [8].

DS3231 - це доступний модуль, який має надзвичайно точний годинник реального часу (RTC) з вбудованою температурною компенсацією кварцового генератора і кристала. Цей модуль оснащений літієвою батареєю, яка забезпечує безперебійне живлення навіть у разі відключення основного джерела живлення.

Інтегрований генератор додатково покращує точність пристрою і дозволяє зменшити кількість потрібних компонентів для його функціонування.

Технічні параметри:

- 1) напруга живлення: 3.3В і 5В;
- 2) чіп пам'яті: АТ24С32 (32 Кб);
- 3) точність:  $\pm 0.432$  сек в день;
- 4) частота кварцу: 32.768 кГц;
- 5) підтримуваний протокол: I2C;
- 6) габарити: 38мм x 22мм x 15мм.

У мікросхемі DS3231, що використовується в цьому модулі, вбудовано кварцовий генератор і датчик температури. Датчик компенсує зміни температури, щоб забезпечити точність часу (зчитування температури також можливе при потребі). Чіп DS3231 підтримує секунди, хвилини, години, день тижня, дату, місяць і рік, а також враховує кількість днів у місяці та робить поправку на високосний рік. Він також підтримує два формати годинника - 24-годинний і 12-годинний, і має можливість програмування двох будильників. Модуль працює за допомогою двопровідної шини I2C.

Сам модуль має мікросхему DS3231N. У середині розташована резисторна збірка RP1 (4.7 кОм), яка необхідна для використання кількох модулів з I2C шиною. Є ще одна збірка резисторів, необхідних для підтяжки ліній A0, A1 і A2, які використовуються для зміни адреси мікросхеми пам'яті АТ24С32N. Резистор R5 і діод D1 використовуються для підзарядки батареї. Також на модулі присутня мікросхема пам'яті АТ24С32N, яка не обов'язкова для роботи годинника RTC DS3231N. Резистор R1 і світлодіод Power вказують на включення модуля. Модуль працює через шину I2C, і для зручності ці шини виведені на два роз'єми J1 і J2. Інші контакти можна переглянути нижче. Призначення J1:

- 1) 32K: вихід, частота 32 кГц;
- 2) SQW: вихід;
- 3) SCL: лінія тактирования (Serial CLock);
- 4) SDA: лінія даних (Serial Dfata);
- 5) VCC: «+» живлення модуля;

б) GND: «-» живлення модуля.

Призначення J2:

- 1) SCL: лінія тактирування (Serial CLock);
- 2) SDA: лінія даних (Serial Data);
- 3) VCC: «+» живлення модуля;
- 4) GND: «-» живлення модуля.

Данна мікросхема має 64 байти пам'яті. Перші вісім байт - робочі. У них зберігається час, дата, день тижня. Решта виділені під потреби користувача. У них можна зберігати наприклад які-небудь налаштування. Коли резервне живлення пропадає, вся інформація в цій пам'яті руйнується. Вся робота з годинником (читання та встановлення часу / дати) зводиться до того, щоб читати і записувати потрібні комірки пам'яті.

На рисунку 2.3 зображено мікросхему DS3231[8] та назначення її виводів.

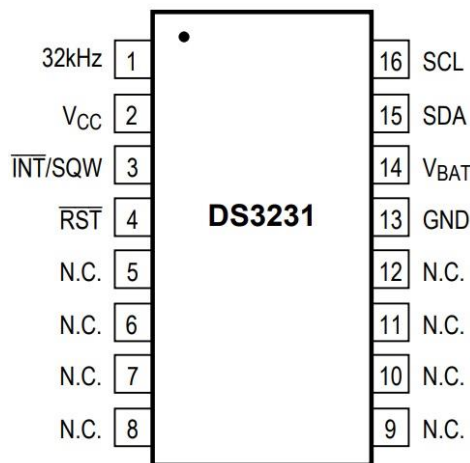


Рисунок 2.3 – Зображення мікросхеми та її виводів

Всі числа в пам'яті зберігаються в двійково-десятковому форматі. Це означає що в одному байті може зберігатися відразу дві цифри. Наприклад число  $0 \times 23$  - містить в собі цифру 2 і цифру 3. На кожен цифру виділяється по 4 біта. Це зроблено для зручності та економії пам'яті. Крім часу і дати в пам'яті зберігаються кілька бітів налаштувань. Clock Halt – керує годинником. Коли біт встановлений то годинник стоїть. Щоб запустити хід годинника необхідно записати в цей біт 0. Після підключення батареї резервного живлення, цей біт установлений і годинник не рахує час.

24 / 12 - це біт вибору режиму годин. Коли цей біт дорівнює одиниці то використовується 12 -ти годинний режим. В іншому випадку 24 -х годинний. Якщо використовується 12 -ти годинний режим то п'ятий біт показує ранок чи вечір зараз. Якщо біт дорівнює 1 то значить зараз вечір. У 24 -х годинному режимі цей біт використовується для зберігання десятків годин спільно з бітом

#### Основні характеристики модуля DS3231:

- 1) годинник реального часу з підрахунком секунд, хвилин, годин, днів, місяця та року (з коригуванням дат в т.ч. по високосному році до 2100 року);
- 2) похибка ходу:  $\pm 2$  хвилини в рік;
- 3) температурний датчик з похибкою  $\pm 3$  ° C;
- 4) два будильника.

#### Технічні характеристики:

- 1) робоча температура:  $-40$  ° C...+  $85$  ° C;
- 2) напруга живлення: 2,3 - 5,5 В;
- 3) напруга живлення батареї: 2,3 - 5,5 В;
- 4) максимальне споживання струму: 650 нА.

#### Плюси використання:

- 5) висока точність ходу годинника;
- 6) широкий діапазон робочої температури.

Схему модуля реального часу DS3231 зображено на рисунку 2.4.

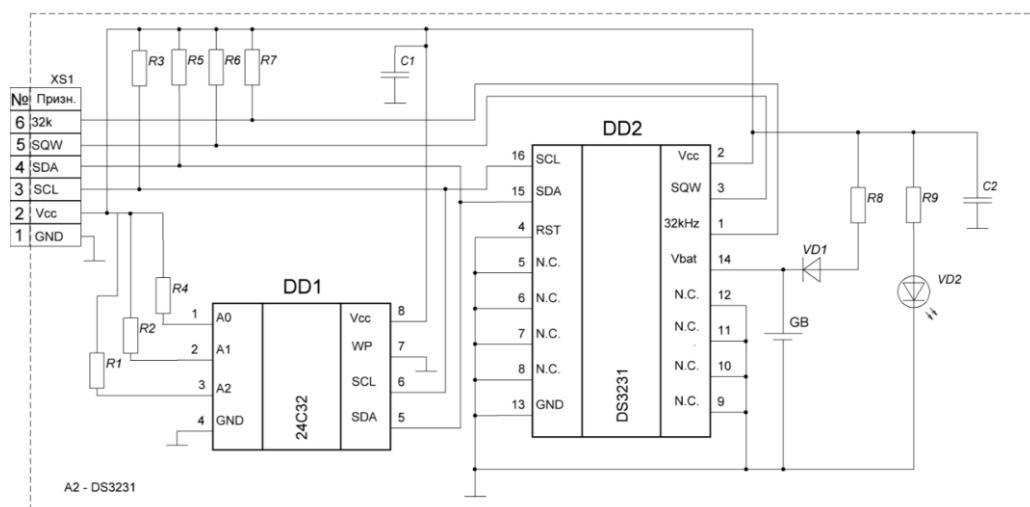


Рисунок 2.4 – Схема модулю реального часу DS3231

Приклад підключення модулю DS3231 до модулю Arduino Nano зображено на рисунку 2.5.

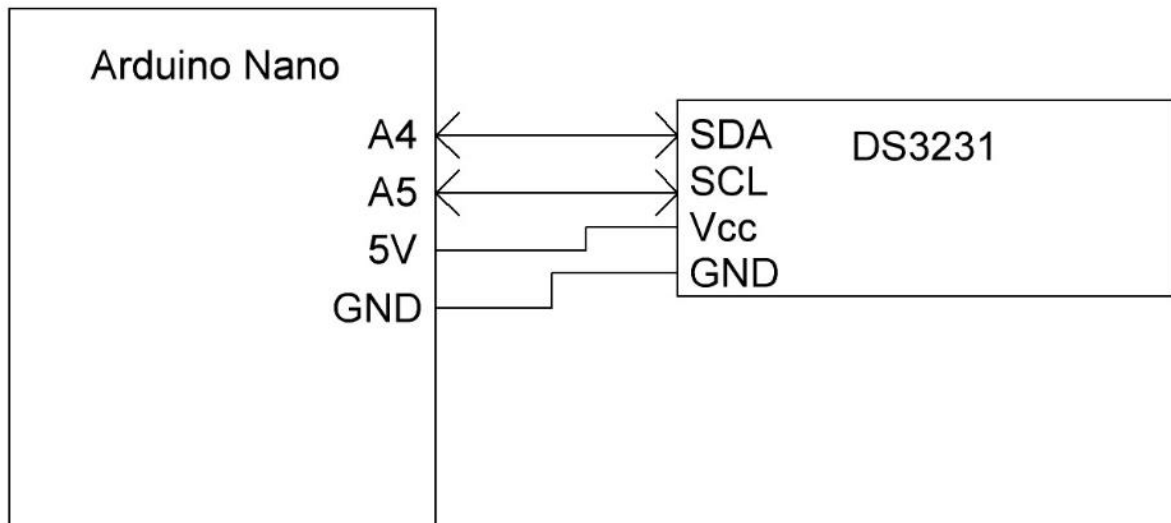


Рисунок 2.5 – Схема підключення модулю реального часу до Arduino Nano

## 2.2 Вибір елементів індикації та схеми керування

В якості елементів індикації будуть використовуватись семисегментні модулі реалізовані на сервоприводах SG-90 [9], які будуть підіймати та опускати окремі сегменти модуля.

Схема індикації зображена на рисунку 2.6 та наведена в Додатку Б.

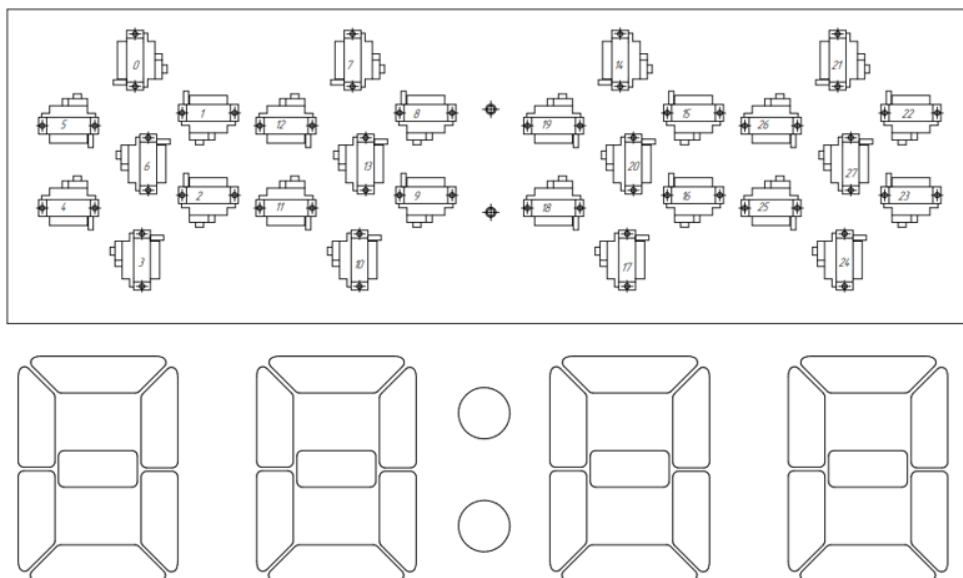


Рисунок 2.6 – Схема індикації

Сервопривід складається з:

- 1) потенціометра;
- 2) двигуна постійного струму;
- 3) вихідного валу;
- 4) шестерні редуктора;
- 5) плати управління, на яку подається керуючий сигнал.

Характеристики: [19]

- 1) швидкість без навантаження: 0.12 сек/60 град. при живленні 4.8В;
- 2) обертаючий момент: 2 кг/см;
- 3) температурний режим: -30 to +60°C;
- 4) робоча напруга живлення: 3.5-5 В;
- 5) струм в русі: 50-80 мА;
- 6) кут повороту 180 град.

На рисунку 2.7 зображена схема модуля керування сервоприводами.

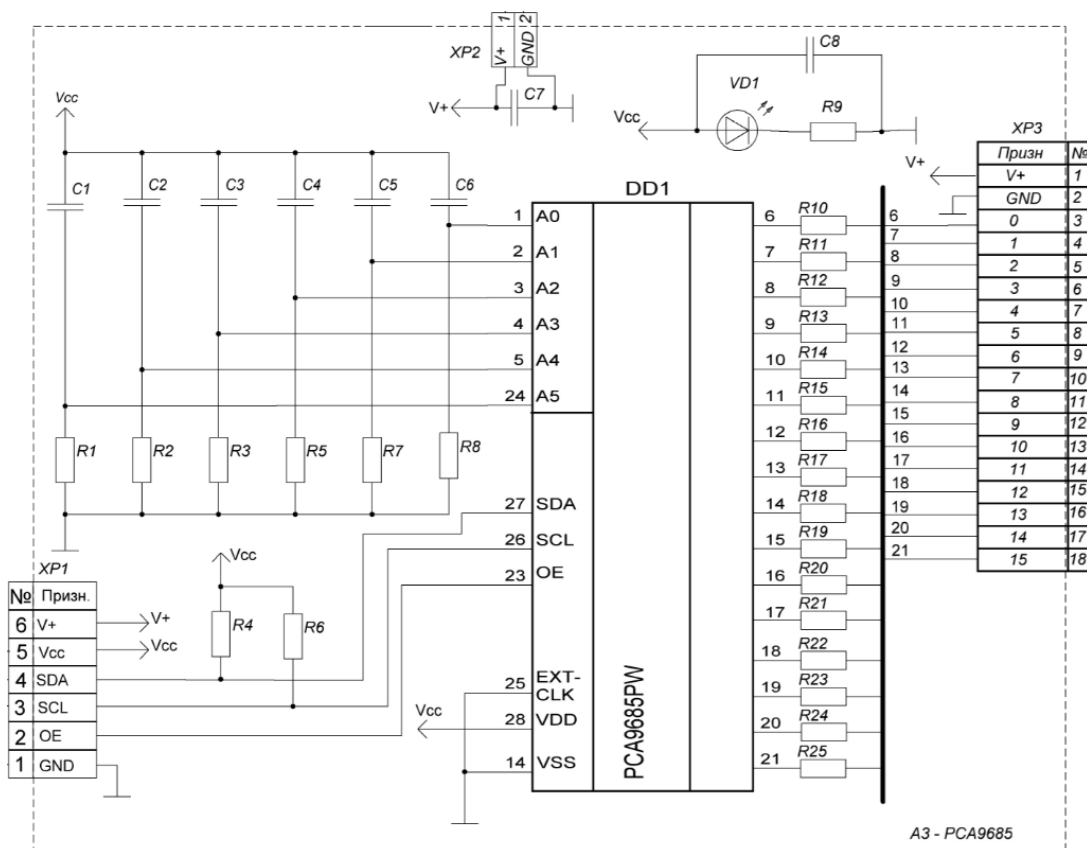


Рисунок 2.7 – Схема модуля керування PCA9685

Через те що в радіотехнічній системі відображення інформації на сервоприводах використовується чотири семисегментних модуля, то кількість сервоприводів дорівнює 28, і через це на платі Arduino Nano не вистачає виходів для їх керування, тому використовується два модуля керування сервоприводами PCA9685 [10] по одному на 2 сегмента його реалізація зображена на рисунку 2.8.

Після вирішення проблеми з керуванням, потрібно визначити яким чином буде відбуватись ідикація за допомогою сервопривода, для цього можна розробити 3Д модель сегментів які будуть підніматись та опускатись сервоприводами, а також потрібно визначитись з розміщенням сервоприводів та загальним виглядом панелі індикації.

Наступним кроком буде проведено розробку функціональної схеми радіотехнічної системи відображення інформації на сервоприводах.

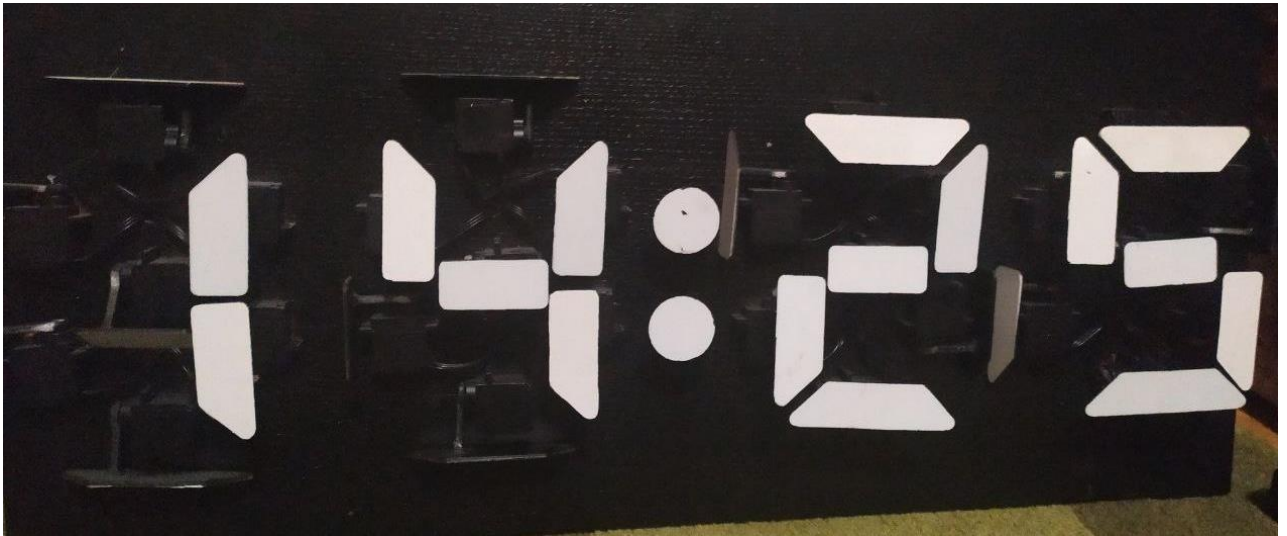


Рисунок 2.8 – Зібраний модуль радіотехнічної системи відображення інформації на сервоприводах

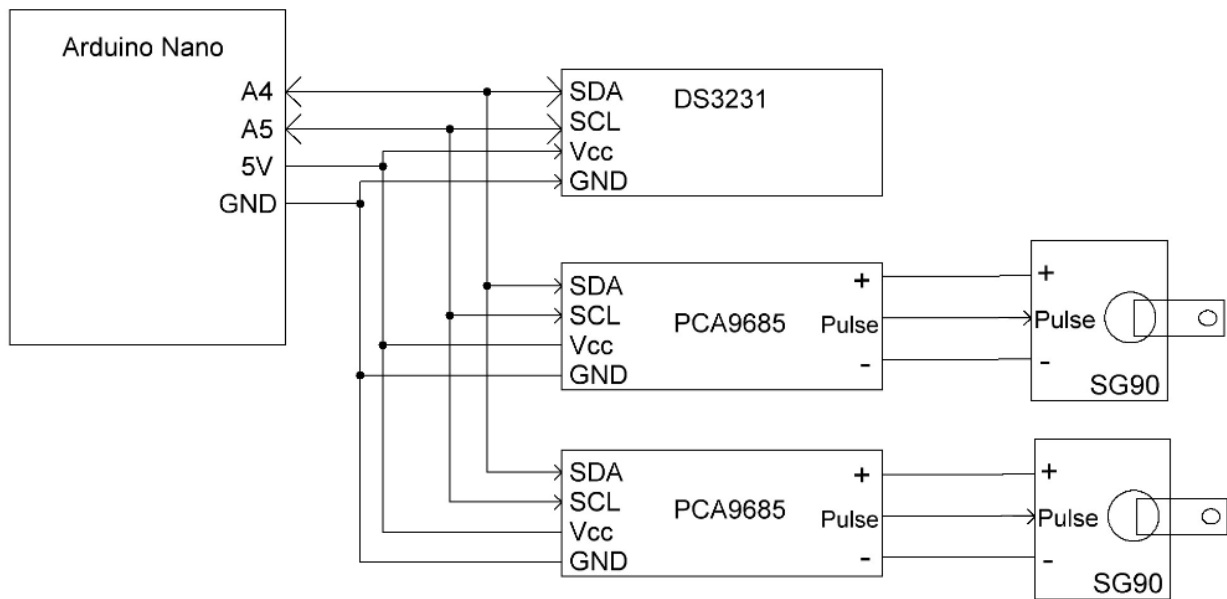


Рисунок 2.9 – Функціональна схема радіотехнічної системи відображення інформації на сервоприводах

У схемі керування подається зовнішнє яке становить 5В. Застосовані роз'єми A4, A5 для зчитування даних з модуля реального часу та подачі сигналу на драйвери керування сервоприводами. Всі підключені модулі до цих роз'ємів працюють по шині I2C. Функціональна схема радіотехнічної системи відображення інформації на сервоприводах зображена на рисунку 2.9 та Додатку Б.

### 2.3 Розробка програмного забезпечення

Розробка алгоритму роботи програми:

Модуль Arduino Nano працює з датчиком реального часу та драйверами керування сервоприводів через роз'єми A4, A5, та формує шину I2C. При запуску програма опускає а згодом піднімає всі сегменти та дає можливість відредагувати верхнє положення планки за умови, що модуль Arduino підключений до комп'ютера через USB. Далі встановлюється час з RTC модуля та оновлюється кожні 60 секунд

```
void timeTick() {
```



```

if (clock_state) {
  if (millis() - timeTimer > 60000) { // відраховуємо хвилину
    timeTimer = millis();
    // кожен хвилину беремо час з RTC
    DateTime now = rtc.now();
  }
}

```

При припиненні подачі живлення на прилад сегменти залишаються в останньому зайнятому положенні.

Також через те що формуються семисегментні модулі з сервоприводів є можливість застосувати бібліотеку для семисегментних індикаторів, ще використовується декілька команд для того, щоб сегменти піднімалися плавно без різких рухів.

Лістинг програми [11] для налаштування приладу через допоміжне меню «Arduino IDE»

```

void help() {
  Serial.println(F("*****"));
  Serial.println(F("Welcome to ServoClock! You can use serial
  commands:"));
  Serial.println(F("<set-time HH:MM> - set time (example: set-time
  01:20)"));
  Serial.println(F("<get-time> - return RTC time"));
  Serial.println(F("<start-clock> - start counting time"));
  Serial.println(F("<stop-clock> - stop counting time"));
  Serial.println(F("<start-string> - start running string"));
  Serial.println(F("<stop-string> - stop running string"));
  Serial.println(F("<all-down> - turn all segments down"));
  Serial.println(F("<all-up> - turn all segments up"));
  Serial.println(F("<down NUMBER> - turn segment NUMBER down
  (example: down 10)"));
  Serial.println(F("<up NUMBER> - turn segment NUMBER up (example:
  up 10)"));
  Serial.println(F("<speed VALUE> - set servo speed to VALUE,
  default 15 (example: speed 20)"));
  Serial.println(F("<adj NUMBER VALUE> - set max VALUE for servo
  with NUMBER and store it in EEPROM (example: adj 05 300)"));
  Serial.println(F("<help> - print this instruction again"));
  Serial.println(F("*****
  *****"));
}

```

Далі буде обрано та налаштовано середовище розробки.

Для розробки програмного забезпечення для Arduino Nano було обрано програмне середовище [12] Arduino IDE

Розглянемо процес створення програми у середовищі Arduino IDE. Запустимо програму Arduino IDE. Для створення проекту потрібно вибрати Файл> Новий. Зображено на рисунку 2.10

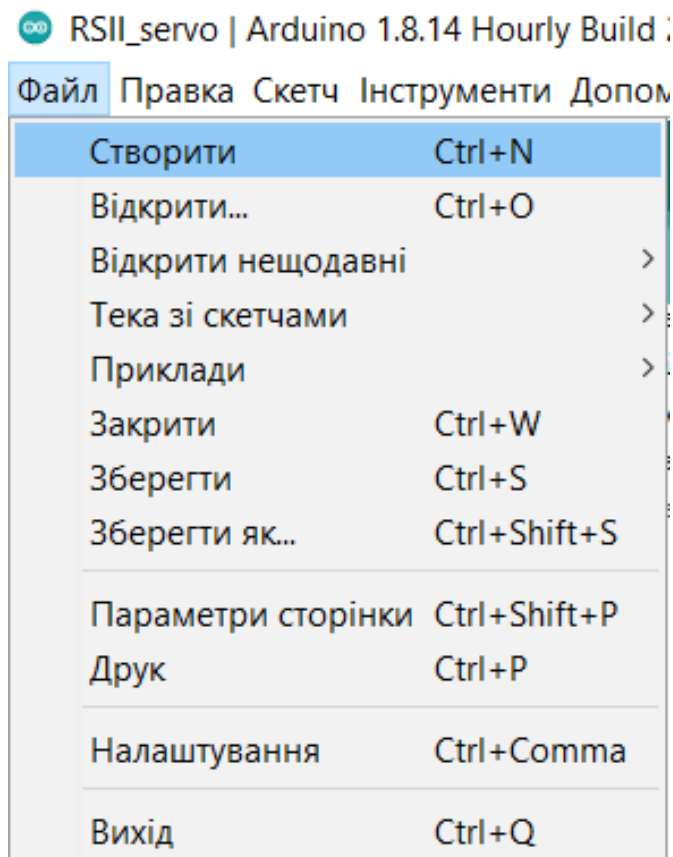


Рисунок 2.10 – Вікно провідника створення проектів

У вікні, що з'явиться вибираємо тип плати зображений на рисунку 2.11. У випадяючому списку Інструменти>Плата вибираємо Arduino Nano.

Так як в схемі використовується модуль-датчик, DS3231. То потрібно реалізувати різні функції [13], то це все потребує установки спеціальних бібліотек для коректної роботи.

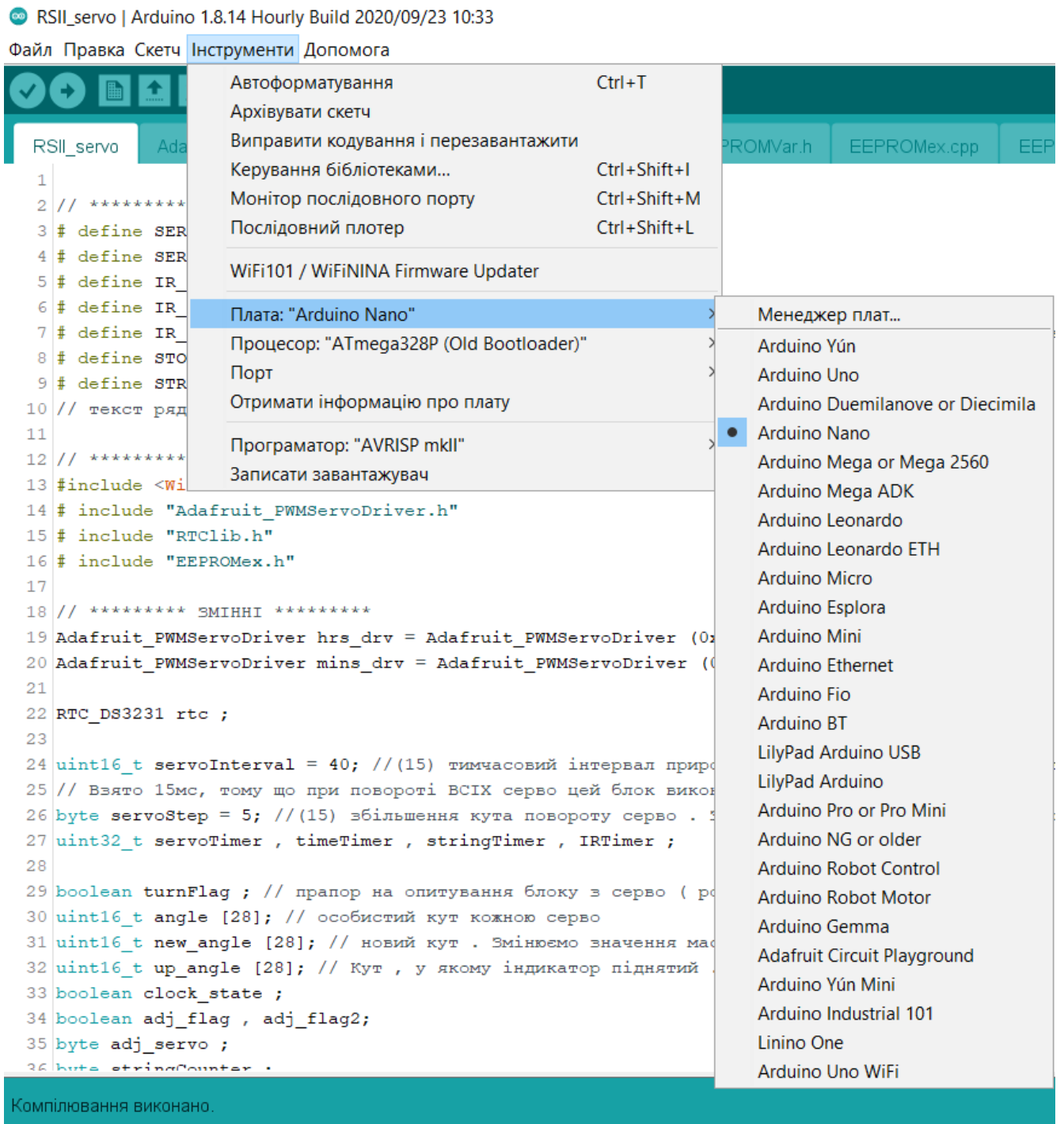


Рисунок 2.11 – Вибір типу плати

Після проведення налаштування можна приступати до написання так званого скетчу, що зображено на рисунку 2.12.

```

1
2 // ***** НАЛАШТУВАННЯ *****
3 # define SERVOMIN 150 // мінімальний сигнал серво
4 # define SERVOMAX 550 // максимальний сигнал серво
5 # define IR_PIN 2 // пін ІЧ датчика
6 # define IR_SENSOR 1 // використовувати ІЧ датчик руху
7 # define IR_TIMEOUT 3000000000000000000 // таймот датчика руху ( замовч . 300000мс це 5 хв )
8 # define STOCK_UP 270 // значення максимального кута (сегмент піднятий ) за умовчанням
9 # define STRING_SPEED 500 // швидкість біжучого рядка
10 // текст рядка , що біжить , налаштується приблизно у рядку №100
11
12 // ***** БІБЛІОТЕКИ *****
13 #include <Wire.h>
14 # include "Adafruit_PWMServoDriver.h"
15 # include "RTCLib.h"
16 # include "EEPROMex.h"
17
18 // ***** ЗМІННІ *****
19 Adafruit_PWMServoDriver hrs_drv = Adafruit_PWMServoDriver (0x40);
20 Adafruit_PWMServoDriver mins_drv = Adafruit_PWMServoDriver (0x41);
21
22 RTC_DS3231 rtc ;
23
24 uint16_t servoInterval = 40; //(15) тимчасовий інтервал прирощення кута для плавного повороту .
25 // Взято 15мс, тому що при повороті ВСІХ серво цей блок виконується 18 мс
26 byte servoStep = 5; //(15) збільшення кута повороту серво . Зберігається в EEPROM ( servo speed )
27 uint32_t servoTimer , timeTimer , stringTimer , IRTimer ;
28
29 boolean turnFlag ; // прапор на опитування блоку з серво ( розвантажує процесор )
30 uint16_t angle [28]; // особистий кут кожною серво
31 uint16_t new_angle [28]; // новий кут . Змінюємо значення масиву - серво повертається
32 uint16_t up_angle [28]; // Кут , у якому індикатор піднятий . Зберігається в EEPROM
33 boolean clock_state ;
34 boolean adj_flag , adj_flag2;
35 byte adj_servo ;
36 byte stringCounter .

```

Рисунок 2.12 – Кінцевий лістинг програми

Бібліотека - це набір функцій, призначених для того, щоб максимально спростити роботу з різними датчиками, ЖК-екранами, модулями та ін. Наприклад, вбудована бібліотека Wire для взаємодії по лінії I2C. Існують сотні додаткових бібліотек, які можна скачати в Інтернеті. Стандартні бібліотеки Arduino і ряд найбільш часто використовуваних додаткових бібліотек перераховані в довідці. Але перед тим, як використовувати додаткові бібліотеки, необхідно спершу встановити їх.

Для встановлення бібліотек потрібно у спливаючому списку Скетч>Додати бібліотеку потрібно обрати нам бібліотеку для реалізації

конкретної функції або для роботи конкретно модуля-датчика, що зображено на рисунку 2.13.

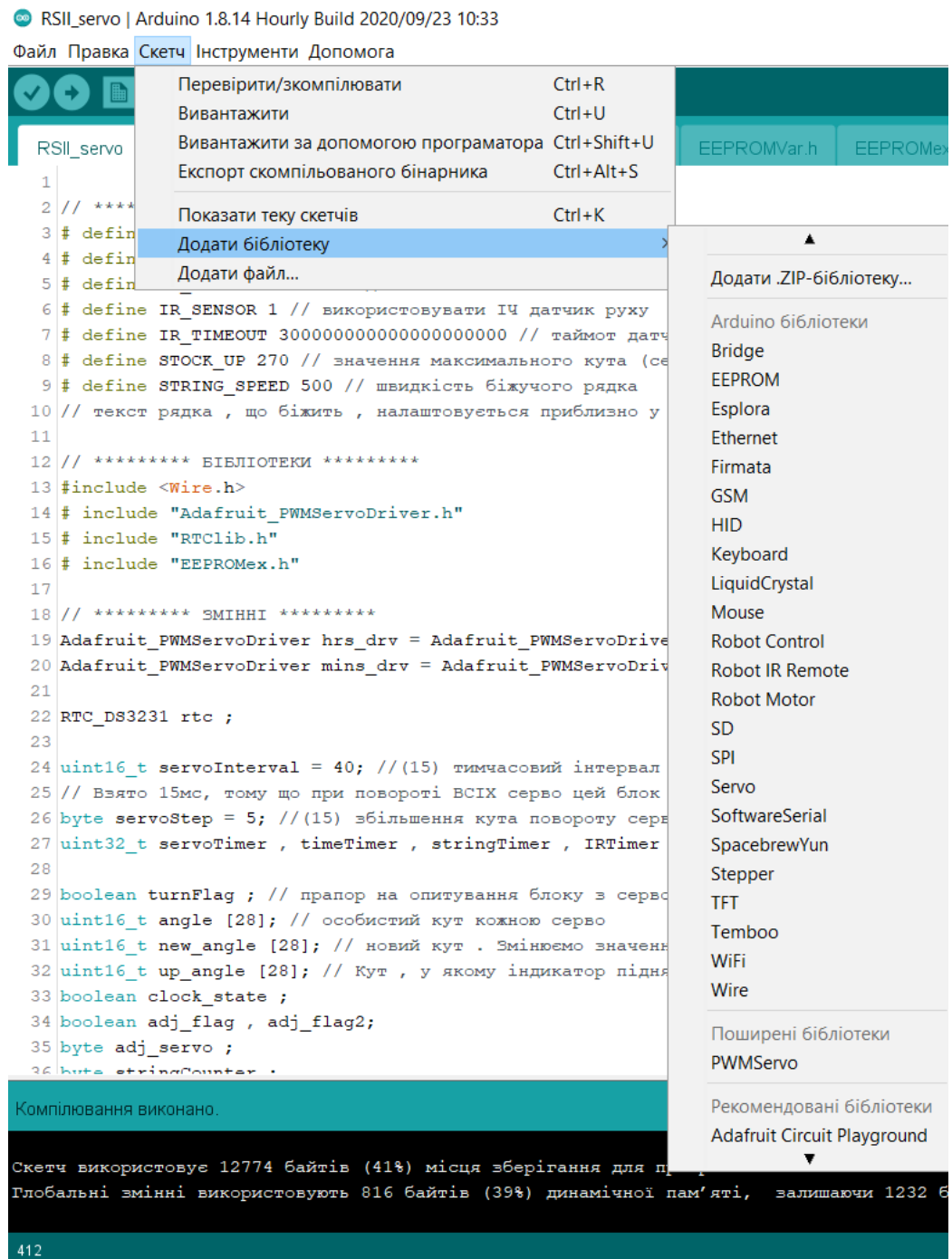


Рисунок 2.13 – Процес встановлення бібліотеки

Так як всі потрібні нам бібліотеки було попередньо скачано з глобальної мережі Інтернет, вони були вставлені в папку з скетчком та задані таким чином, а саме:

- 1) "Adafruit\_PWMServoDriver.h";

- 2) "RTClib.h";
- 3) "EEPROMex.h".

Потім потрібно скопіювати прошивку, для цього обрати у спливаючому списку Скетч>Перевірити/Компілювати, в разі вдалої компіляції побачимо, що написано «Компіляція завершена» та відповідну інформацію про це в вікні Arduino IDE, яке зображено на рисунку 2.14.

```

393 else if ( buf.startsWith (" adj ")) { // adj 12 600
394   byte num = buf.substring (4, 6). toInt ();
395   int val = buf.substring (7). toInt ();
396   up_angle [ num ] = val ; // встановити нове значення
397   new_angle [ num ] = SERVOMIN;
398   EEPROM.writeByte ( num * 2, val ); // записати на згадку
399   Serial.print (F(" Adjust servo #")); Serial.print ( num );
400   Serial.print (F(" value ")); Serial.println ( val );
401   turnFlag = true ;
402   adj_flag = true ;
403   adj_servo = num ;
404 }
405 if ( buf.startsWith (" start-clock ")) {
406   Serial.println (F(" Clock start "));
407   clock_state = true ;
408 }
409 if ( buf.startsWith (" stop-clock ")) {
410   Serial.println (F(" Clock stop "));
411   clock_state = false ;
412 }
413 if ( buf.startsWith (" start-string ")) {
414   Serial.println (F(" Start string "));
415   clock_state = false ;
416   stringFlag = true ;
417 }
418 if ( buf.startsWith (" stop-string ")) {
419   Serial.println (F(" Stop string "));
420   clock_state = true ;
421   stringFlag = false ;
422 }
423 if ( buf.startsWith (" help ")) {
424   help ();
425 }
426 }
427 }
428

```

Компілювання виконано.

Скетч використовує 12774 байтів (41%) місця зберігання для програм. Межа 30720 байтів.  
Глобальні змінні використовують 816 байтів (39%) динамічної пам'яті, залишаючи 1232 байтів для локальних змінних. Межа 2048 байтів.

Рисунок 2.14 – Вікно Arduino IDE з завершеною компіляцією

Коли виконали успішну компіляцію, підключаємо плату Arduino Nano до ПК та у випадяючому вікні Інструменти>Порт обираємо порт по якому у нас підключена плата та нажимаємо «Загрузка» в пункті меню «Скетч», після цього почнеться загрузка прошивки в нашу Arduino Nano. Після успішної загрузки

побачимо відповідну інформацію в вікні Arduino IDE, що зображено на рисунку 2.15.

Програма написана на мові C/C++ в програмному середовищі Arduino IDE.

Лістинг програми наведено в додатку В.

```

RSII_servo | Arduino 1.8.14 Hourly Build 2020/09/23 10:33
Файл Правка Скетч Інструменти Допомога

RSII_servo Adafruit_PWMServoDriver.cpp Adafruit_PWMServoDriver.h EEPROMVar.h EEPROMMex.cpp EEPROMMex.h RTCLib.cpp RTCLib.h

172 Serial.println(F("*****"));
173 Serial.println (F("< set-time HH:MM> - set time ( example : set-time 01:20)"));
174 Serial.println (F("< get-time > - return RTC time "));
175 Serial.println (F("<start-clock> - start counting time "));
176 Serial.println (F("< stop-clock > - stop counting time "));
177 Serial.println (F("< start-string > - start running string "));
178 Serial.println (F("< stop-string > - stop running string "));
179 Serial.println (F("< all-down > - turn all segments down "));
180 Serial.println (F("< all-up > - turn all segments up "));
181 Serial.println (F("< down NUMBER> - turn segment NUMBER down ( example : down 10)"));
182 Serial.println (F("< up NUMBER> - turn segment NUMBER up ( example : up 10)"));
183 Serial.println (F("< speed VALUE> - set servo Speed to VALUE, default 15 ( example : speed 20)"));
184 Serial.println (F("< adj NUMBER VALUE> - set max VALUE for servo with NUMBER and store it in EEPROM ( example : adj 05300)"));
185 Serial.println (F("< help > - print this instruction again "));
186 Serial.println(F("*****"));
187 }
188
189 void loop () {
190   serialTick (); // обробка команд з порту
191   servoTick (); // плавне управління серво
192   timeTick (); // таймер часу ( прапором clock_state )
193   displayString (); // відобразити рядок , що біжить (по прапору stringFlag )
194   IRTick (); // Перевірка ІЧ датчика
195 }
196 void IRTick () {
197
198   IRFlag = true ;
199
200   // if (IR_SENSOR) {
201   //   if ( digitalRead (IR_PIN)) { // якщо спрацював датчик руху
202   //     IRTimer = millis (); // скинути таймер
203   //     if (! IRFlag ) IRFlag = true ; // увімкнути годинник
204   //   } else { // якщо датчик показує 0
205   //     IRFlag = false ;
206   //   }
207   // }

Компілювання виконано.

Скетч використовує 12774 байтів (41%) місця зберігання для програм. Межа 30720 байтів.
Глобальні змінні використовують 816 байтів (39%) динамічної пам'яті, залишаючи 1232 байтів для локальних змінних. Межа 2048 байтів.

129

```

Рисунок 2.15 – Вікно Arduino IDE з завершеною загрузкою прошивки

Наступним кроком буде розробка програмного забезпечення.

Лістинг першочергових команд та алгоритм запуску системи:

```

void setup () {
  Serial.begin (9600);
  Serial.setTimeout (300);

  if (!IR_SENSOR) IRFlag = true ; // якщо не використовуємо ІЧ, дозволити роботу

  // Перевірка RTC
  if ( rtc.begin ()) {

```

```

    Serial.println ("RTC OK");
    clock_state = true ;
} else {
    Serial.println ("RTC ERROR");
    clock_state = false ;
}
help (); // Виведення текстового меню

// старт і налаштування серво шилда
hrs_drv.begin ();
hrs_drv.setPWMPfreq (60); // Analog servos run at ~60 Hz updates
mins_drv.begin ();
mins_drv.setPWMPfreq (60); // Analog servos run at ~60 Hz updates

// перший запуск
if ( EEPROM.read (100) != 10) { // зберігаємо якесь число як індикатор
запуску
    Serial.println (" first start ");
    EEPROM.writeByte (100, 10);
    for ( byte i = 0; i < 28; i++) {
        EEPROM.writeInt (i * 2, STOCK_UP);
        EEPROM.writeByte (90, 15);
    }
}

servoStep = EEPROM.read (90); // Згадуємо швидкість повороту

// при запуску все серво опустити
for ( int i = 0; i < 28; i++) {
    new_angle [i] = SERVOMIN;
    angle [i] = SERVOMIN;
    up_angle [i] = EEPROM.readInt (i * 2); // відновити з пам'яті верхні
значення індикаторів
    if (i < 14)
        hrs_drv.setPWM (i, 0, angle [i]); // перший шилд

    else {
        mins_drv.setPWM (i - 14, 0, angle [i - 14]); // другий шилд
    }

    delay (120); //(70) затримка між поворотами , ніж брати великий струм
}

delay (700);
for ( int i = 0; i < 28; i++) {
// відключити серво
    if (i < 14)
        hrs_drv.setPWM (i, 0, 4096);
    else {
        mins_drv.setPWM (i - 14, 0, 4096);
    }
}

delay (700);
Serial.println (F(" Servo ready "));
timeTimer = 60000; // Штучно переповнити таймер часу _
// щоб годинник відразу показали час (див. timeTick ())
}

```

**Лістинг для зчитування часу з модуля RTC та виведення часу за допомогою сервоприводів:**



```

void loop() {
  serialTick();      // обробка команд з порта
  servoTick();      // плавне керування серво
  timeTick();       // таймер часу (по прапору clock_state)
}
void timeTick() {
  if (clock_state) {
    if (millis() - timeTimer > 60000) { // відрахуємо хвилину
      timeTimer = millis();
      // кожен хвилину берем час з RTC тип

      DateTime now = rtc.now();
      Serial.print("New minute. Time is ");
      Serial.print(now.hour());           Serial.print(":");
      Serial.println(now.minute());
      // забиваємо масив для відображення
      uint8_t indicators[4];
      indicators[0] = now.hour() / 10;    // десятки годин
      indicators[1] = now.hour() % 10;    // одиниці годин
      indicators[2] = now.minute() / 10;  // десятки хвилин
      indicators[3] = now.minute() % 10;  // одиниці хвилин
      turnFlag = true;                   // дозволити поворот серв
      byte servoNum = 0;
      for (byte i = 0; i < 4; i++) {      // для 4 індикаторів
        for (byte j = 0; j < 7; j++) {    // кожен сегмент
          boolean thisBit;
          // thisBit приймає 0 або 1 - підняти чи опустити сегмент.
          // HEXnumbers розбивається на біти, наприклад 0000110 це цифра 1
          // і побітно приймається рішення, підняти чи опустити сегмент
          thisBit = HEXnumbers[indicators[i]] & (1 << j);
          if (i == 0 && indicators[0] == 0) thisBit = 0;
          // не показувати ноль в десятках годин
          if (thisBit) new_angle[servoNum] = up_angle[servoNum]; // підняти
          else new_angle[servoNum] = SERVOMIN;
          // опустити
          servoNum++;
        }
      }
    }
  }
}

```

У цьому розділі було проведено вибір модулів та елементної бази для радіотехнічної системи відображення інформації на сервоприводах. Було обґрунтовано вибір схеми системи відображення інформації на сервоприводах, а також видів радіотехнічної системи.

Під час обрання модулю реального часу DS3231 було враховано його надзвичайну точність та можливості компенсації змін температури. Цей модуль

має вбудований кварцовий генератор і датчик температури, що забезпечує точність часу та можливість зчитування температури при необхідності.

Також було обрано модулі керування сервоприводів PCA9685 для забезпечення керування рухом сервоприводів. Ці модулі підтримують широтно-імпульсну модуляцію [14] (ШИМ) і дозволяють регулювати обороти двигуна та яскравість світлодіодів. Вибір був обґрунтований їхньою сумісністю з Arduino Nano і здатністю легко збирати складні схеми навісним монтажем.

У цьому розділі також були розроблені схеми підключення модулю реального часу DS3231 та модулів керування сервоприводів PCA9685 до Arduino Nano. Ці схеми забезпечують правильну комунікацію та взаємодію між компонентами системи.

Крім того, була проведена розробка програмного забезпечення для управління радіотехнічною системою відображення інформації на сервоприводах. Це програмне забезпечення дозволить забезпечити потрібні функції, включаючи керування рухом сервоприводів, читання значень з аналогових датчиків, а також реалізацію годинника і збереження часу з використанням модулю DS3231. Розроблене програмне забезпечення забезпечить плавний рух сервоприводів залежно від вхідних сигналів і забезпечить точне відображення інформації на них.

У результаті вибору модулів і розробки схем підключення та програмного забезпечення було створено функціональну радіотехнічну систему відображення інформації на сервоприводах. Ця система має високу точність часу завдяки модулю DS3231 і здатна ефективно керувати рухом сервоприводів для відображення потрібної інформації. Крім того, система може бути легко розширена і модифікована для виконання інших завдань та функцій залежно від потреб користувача.

Отже, розділ, присвячений вибору модулів, елементної бази та програмного забезпечення для радіотехнічної системи відображення інформації на сервоприводах, був успішно завершений, і було розроблено функціональну систему та написаний лістинг програми, яка відповідає поставленим вимогам і має потенціал для подальшого розширення та вдосконалення.

### **3 МОДЕЛЮВАННЯ РАДІОТЕХНІЧНОЇ СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА СЕРВОПРИВОДАХ**

#### **3.1 Вибір моделюючого пакету**

Під час розробки радіотехнічних систем, важливим етапом є моделювання та дослідження їх функціональності перед фізичною реалізацією. Для цього необхідно обрати підходящий моделюючий пакет, який забезпечить необхідну функціональність та зручне інтерфейсне середовище для роботи.

У даному випадку, для моделювання та дослідження радіотехнічної системи відображення інформації на сервоприводах, було обрано програмне середовище Proteus 8. Proteus 8 є популярним та потужним інструментом для електронного моделювання та симуляції, що надає широкий спектр можливостей для розробки та тестування електронних пристроїв.

Proteus 8 має декілька ключових переваг, які зробили його відповідним вибором для моделювання радіотехнічної системи відображення інформації на сервоприводах в даному випадку. Перш за все, Proteus 8 надає можливість моделювати модульні проєкти й симулювати роботу мікроконтролерних модулів таких як Arduino, що дозволяє вірно відтворювати сигнали і процеси та досліджувати їх вплив на систему відображення інформації на сервоприводах. Це важливо для отримання точних результатів та оцінки функціональності системи у реальних умовах.

Другою важливою перевагою Proteus 8 [15] є його здатність моделювати сигнали комунікації модулів I2C. I2C є одним з найпоширеніших протоколів зв'язку між електронними компонентами, і можливість моделювання цього протоколу дозволяє виконувати дослідження та тестування взаємодії між компонентами системи та перевіряти їх правильність та ефективність.

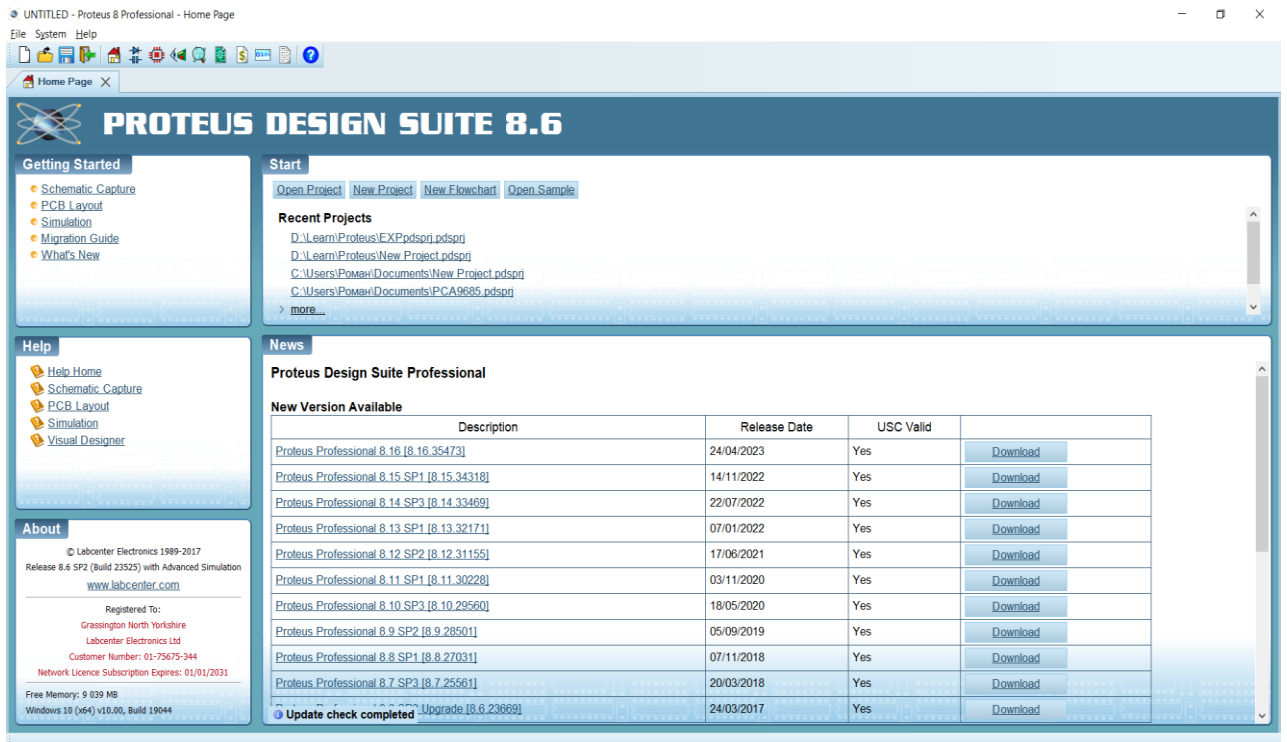


Рисунок 3.1 – Зображення головного меню програмного середовища Proteus 8

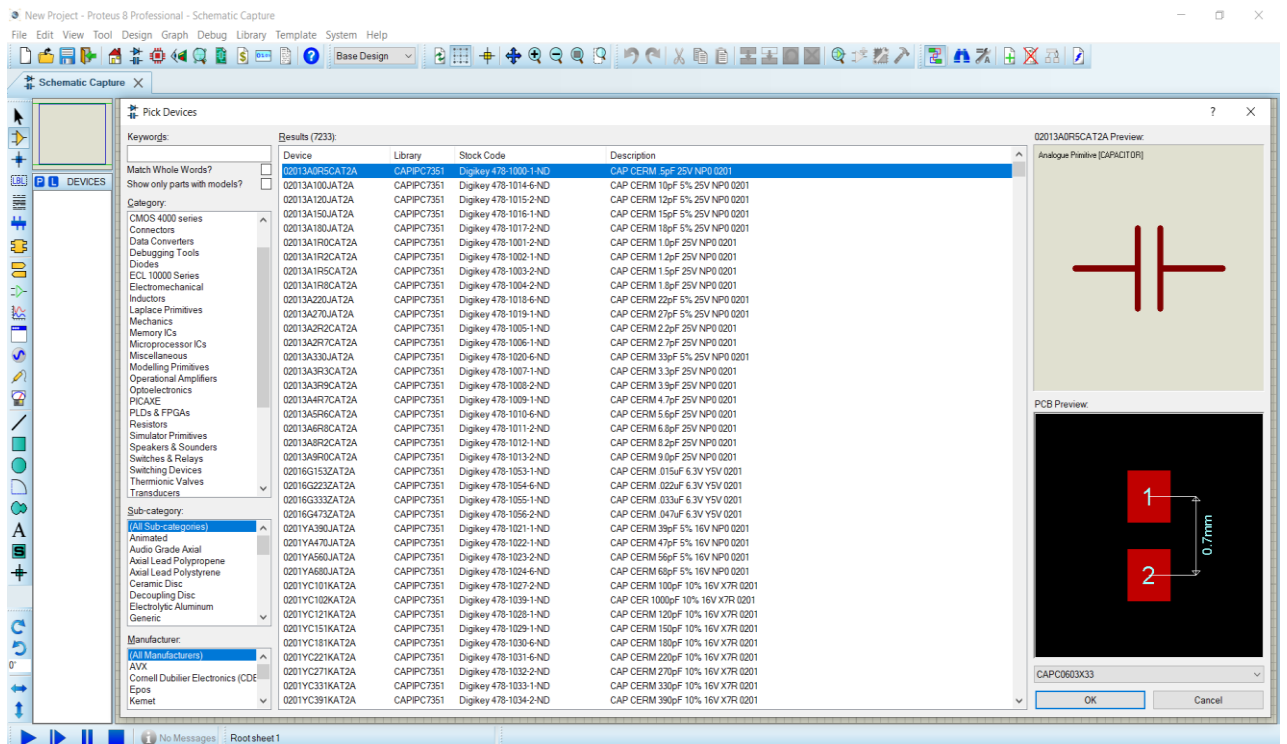


Рисунок 3.2 – Інтерфейс та бібліотека елементів в Proteus 8

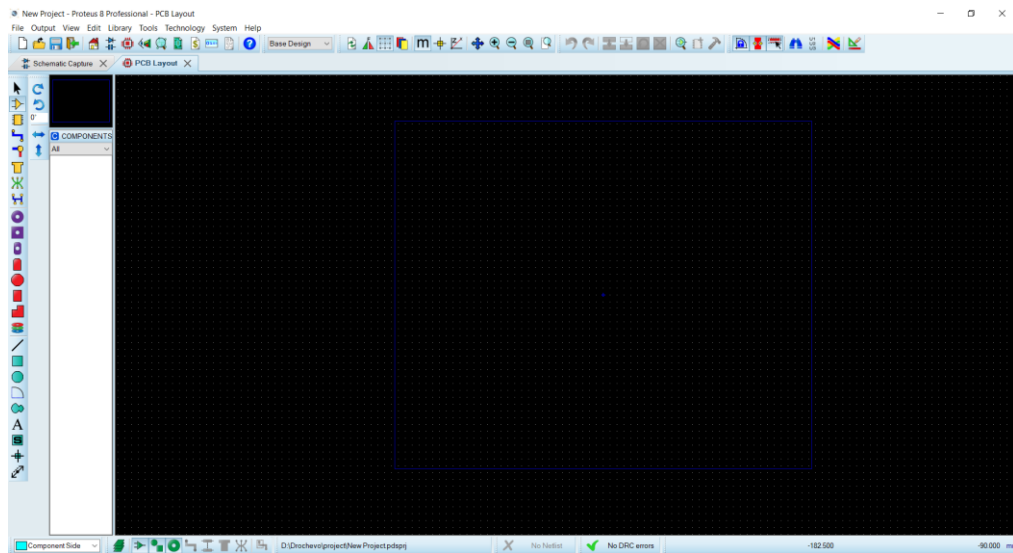


Рисунок 3.3 – Інтерфейс та робоча площа для розробки друкованих плат

Proteus 8 надає зручне інтерфейсне середовище для розробки моделей, що дозволяє з легкістю створювати та налаштовувати елементи системи, встановлювати параметри, здійснювати симуляцію та аналізувати результати. Графічний інтерфейс Proteus 8 є інтуїтивно зрозумілим і дозволяє швидко налаштовувати параметри системи та виконувати симуляцію без складнощів. Головне меню та інтерфейси Proteus 8 зображено на рисунку 3.1-3.3. Нижче наведено зображення програмного середовища Proteus 8, де проводиться моделювання рухомого блоку та сигналів комунікації модулів I2C:

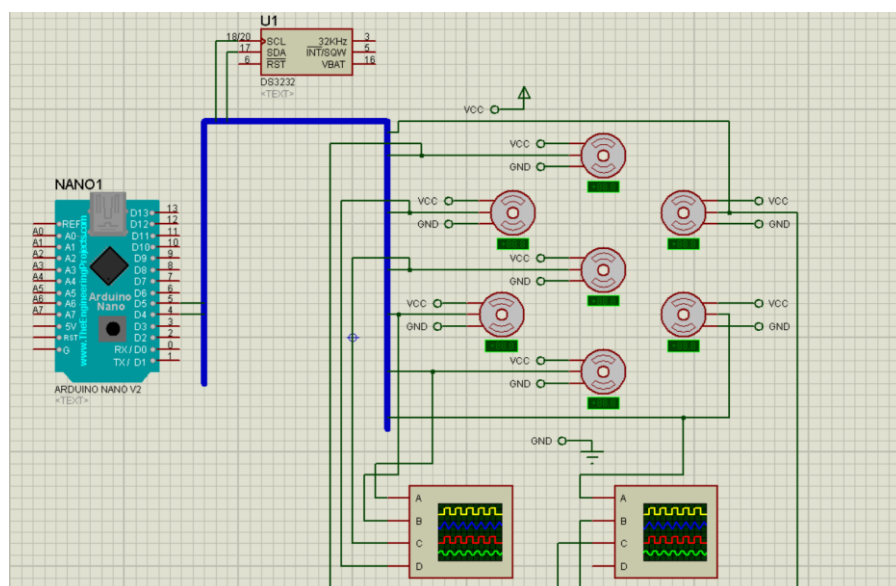


Рисунок 3.4 – Дослідження роботи рухомого модулю радіотехнічної системи відображення інформації на сервоприводах в Proteus 8

Обраний програмний пакет Proteus 8 відповідає потребам моделювання та дослідження Радіотехнічної системи відображення інформації на сервоприводах. Його можливості моделювання рухомого блоку та сигналів комунікації модулів I2C, а також зручне інтерфейсне середовище роботи роблять його відмінним вибором для виконання завдань моделювання та дослідження даної радіотехнічної системи.

### 3.2 Моделювання рухомого блоку радіотехнічної системи відображення інформації на сервоприводах

Моделювання роботи проводилось в два етапи. Спочатку в Arduino IDE була відлагоджена програма RSII\_servo і отриманий файл RSII\_servo.hex, а в пакеті Proteus 8 Professional проводилась перевірка працездатності схеми. Схема радіотехнічної системи відображення інформації на сервоприводах зображено на рисунку 3.4. Викликавши властивості мікроконтролера, зображено на рисунку 3.5, підключаємо отриманий файл RSII\_servo.hex. Після запуску симуляції були зняті покази осцилографа, які зображені на рисунках 3.6-3.8.

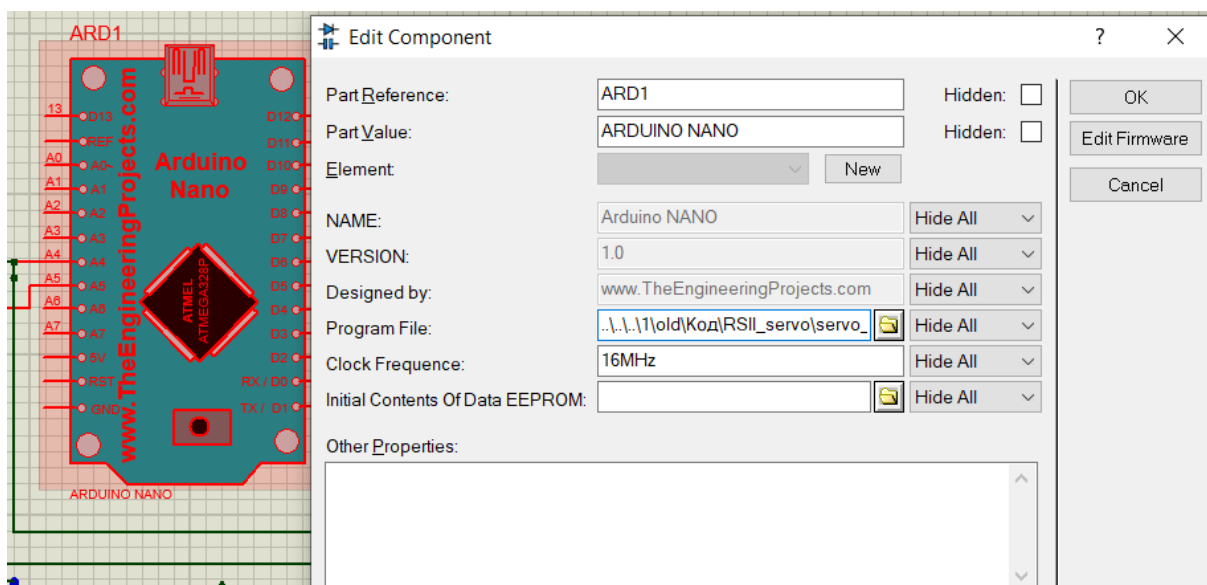


Рисунок 3.5 – Підключення програмного забезпечення RSII\_servo.hex

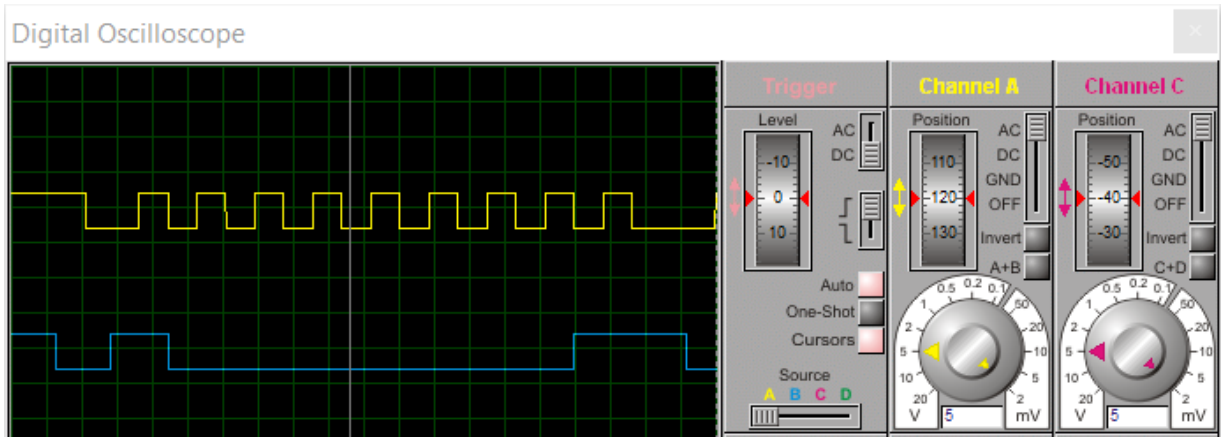


Рисунок 3.6 – Частота запиту даних у RTC модуля

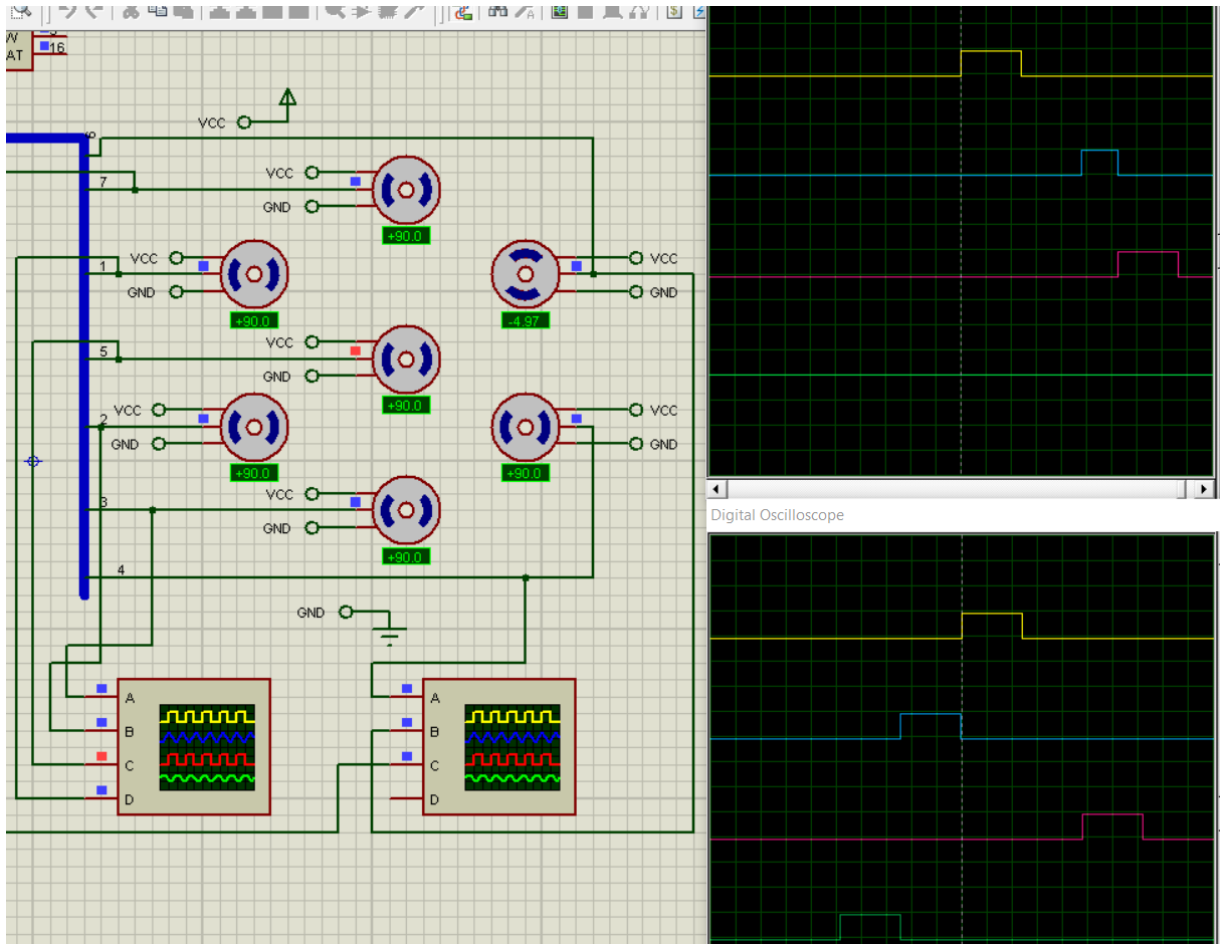


Рисунок 3.7 – Сигнали керування сервоприводів для формування цифри шість

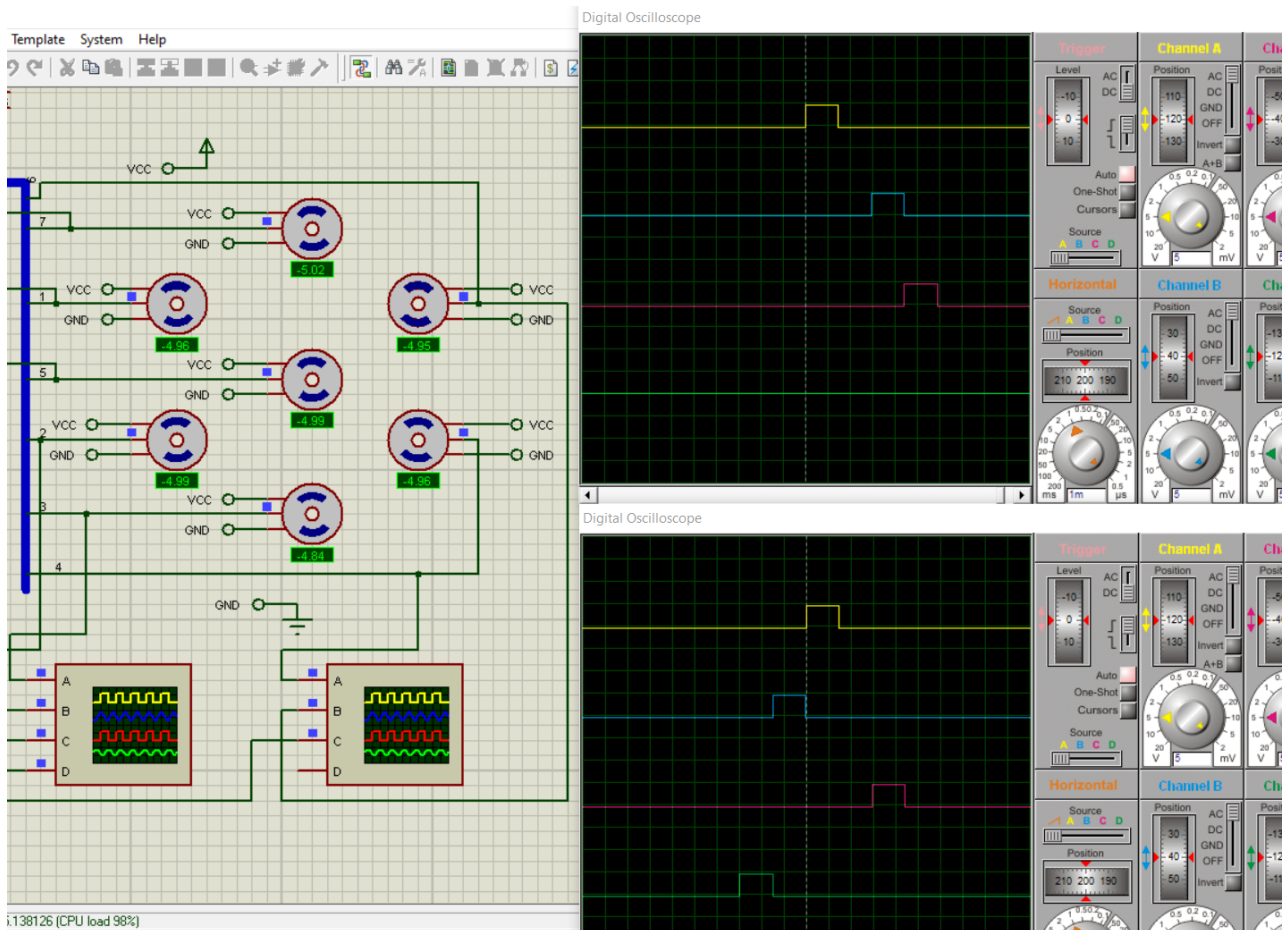


Рисунок 3.8 – Сигнали керування сервоприводів для нульового положення

При проведенні експериментального дослідження в Proteus 8 у годинника одразу йде запит до RTC модуля та команда для сервоприводів зайняти стартове положення. За допомоги осцилографа проводиться дослідження сигналів що виходять з мікроконтролера. Тому в середовищі Proteus можна експериментально дослідити усі сигнали керування і роботу годинника.

### 3.3 Дослідження сигналів комунікації модулів I2C

У сучасних радіотехнічних системах інформація часто передається між різними пристроями за допомогою різних комунікаційних шин. Одна з таких шин - шина I2C (Inter-Integrated Circuit), є популярним засобом зв'язку між пристроями в системах з обмеженими ресурсами.



Цей розділ присвячений дослідженню та аналізу роботи шини [16] I2C за допомогою моделюючого пакету Proteus 8. Ми використали Proteus 8 для збору макету розробляємої радіотехнічної системи, яка включає сервоприводи для відображення інформації його зображено на рисунку 3.6.

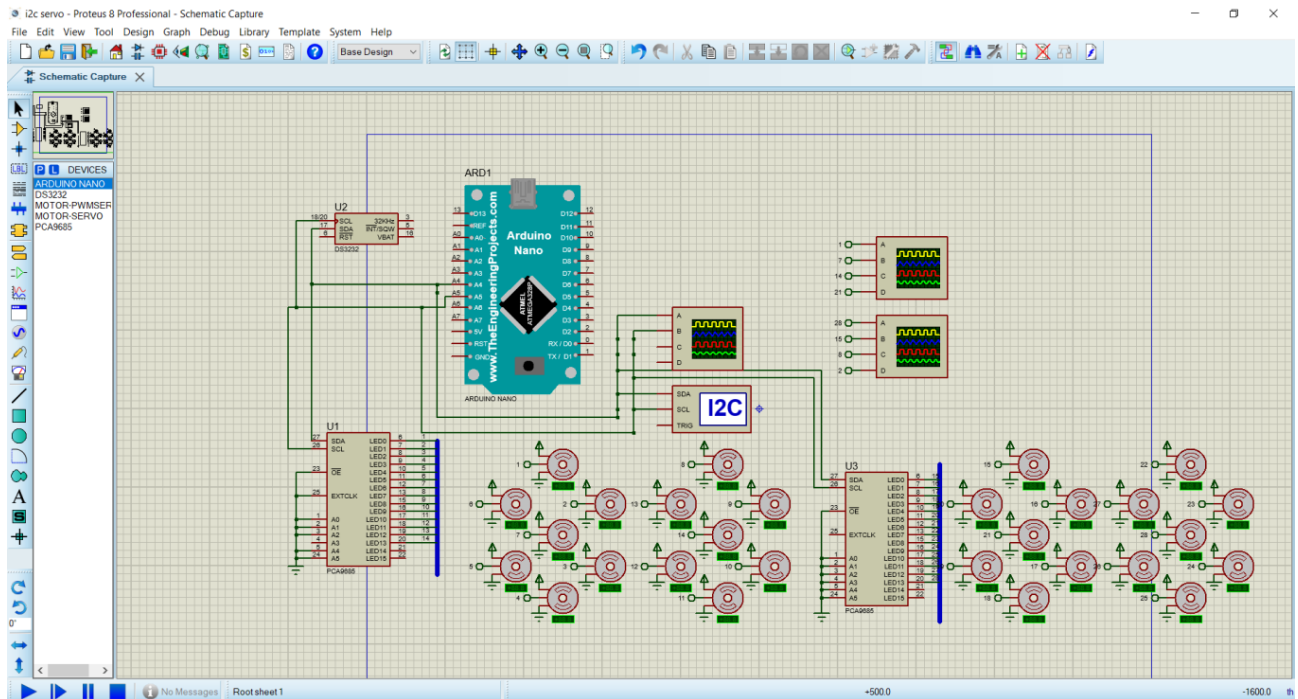


Рисунок 3.9 – Зібраний макет розробляємої радіотехнічної системи

Основна мета дослідження полягала в оцінці швидкості передачі та пропускної здатності шини I2C. Для досягнення цієї мети було розроблено тестовий код, який дозволяє виміряти час передачі даних та визначити швидкість передачі нашої шини I2C результати роботи тестової програми зображено на рисунках 3.10 – 3.12.

Для ініціалізації та коректної роботи шини I2C було змодульовано більшу частину елементів розробляємої радіотехнічної системи. Шиною I2C було з'єднано модуль Arduino Nano та драйвер керування сервоприводами PCA9685 до котрого було під'єднано сервоприводи, задля зручності в моделюючому пакеті з'єднання було реалізовано за допомогою шини.

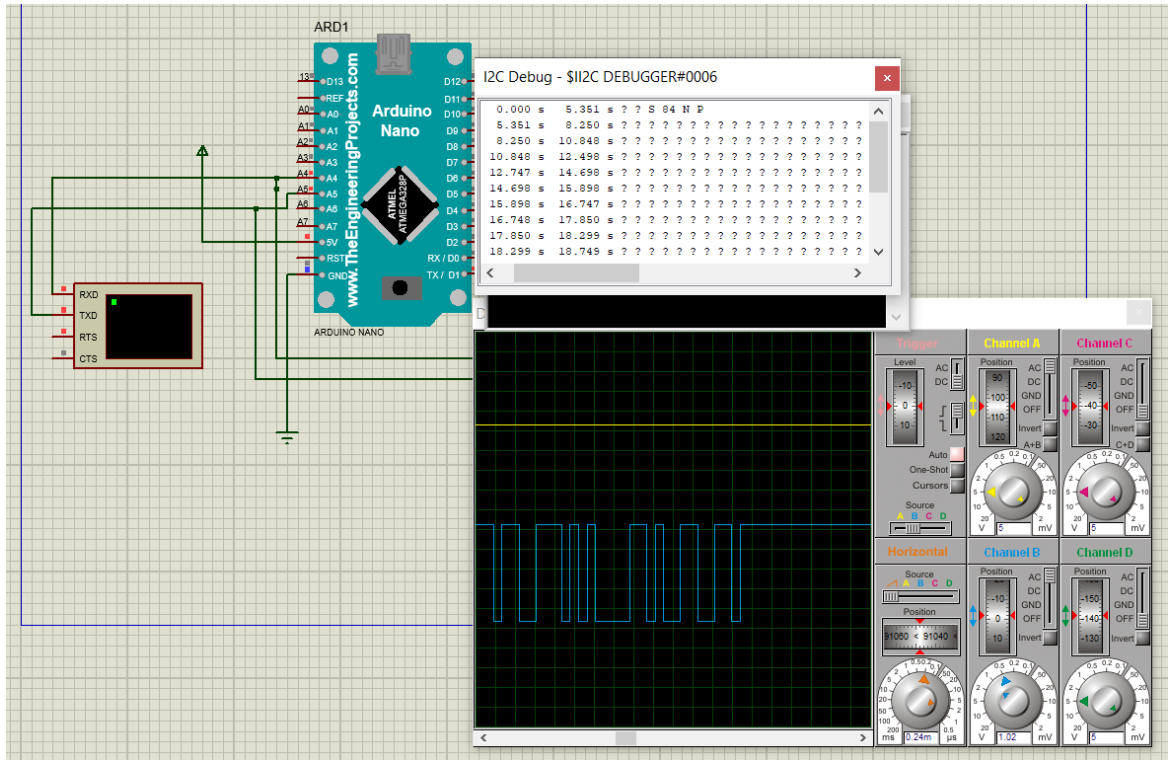


Рисунок 3.10 – Передача тестового повідомлення

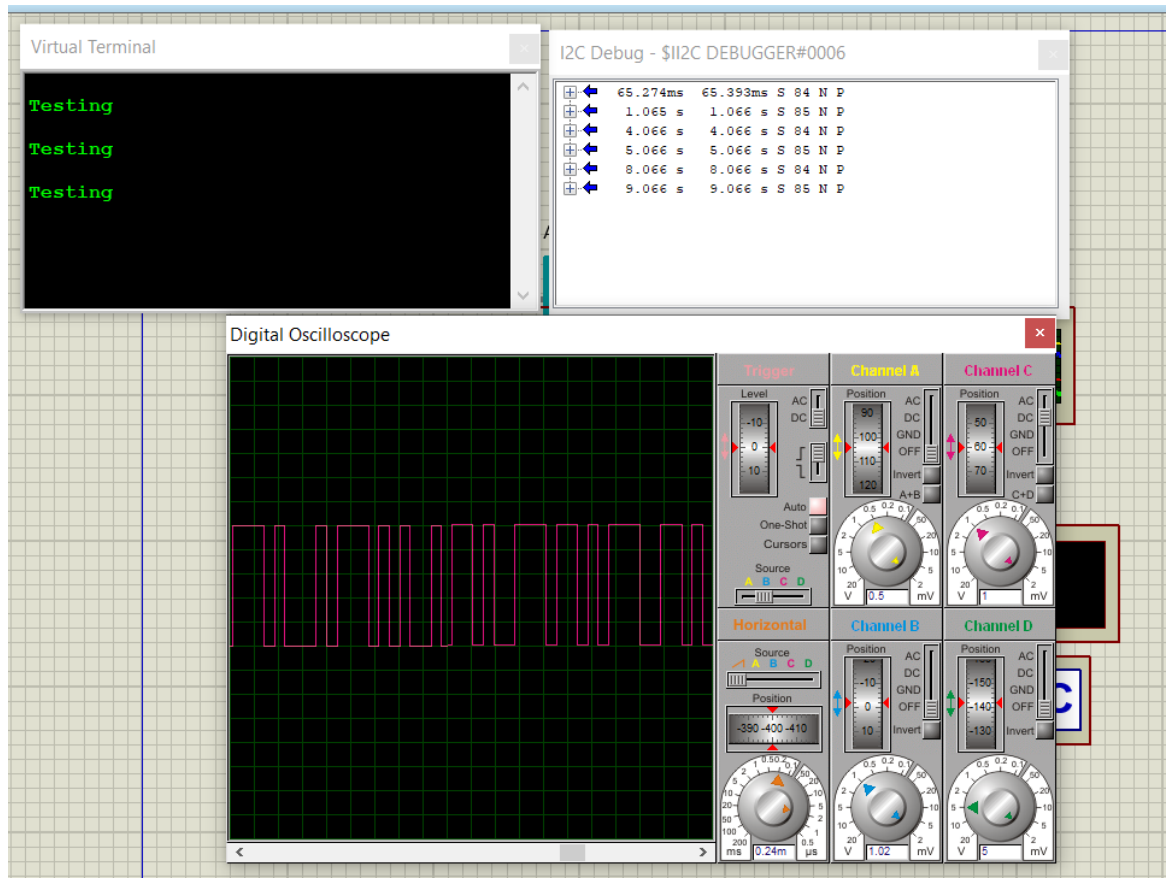


Рисунок 3.11 – Передача текстового повідомлення шиною I2C й відображення результату передачі за допомогою віртуального терміналу

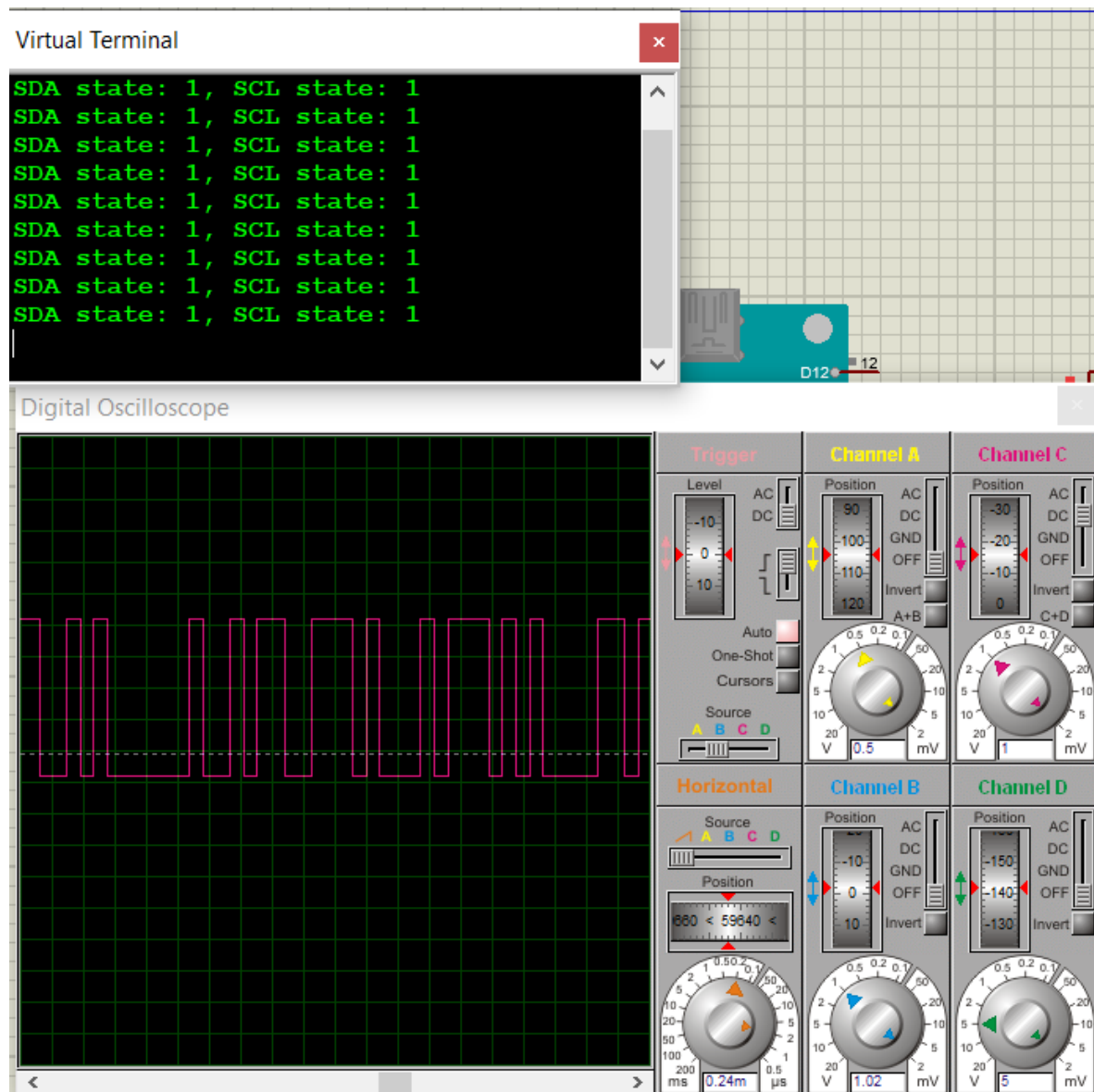


Рисунок 3.12 – Імпульсна послідовність ланцюгу SDA

Процес передачі даних був візуалізований за допомогою осцилографа та терміналу I2C. Це дозволило нам спостерігати сигнали на лініях SDA та SCL та аналізувати їх в реальному часі результати аналізу зображено на рисунках 3.13 та 3.14. За допомогою осцилографа, ми змогли оцінити швидкість передачі даних та виявити можливі проблеми, такі як шуми або спотворення сигналу.

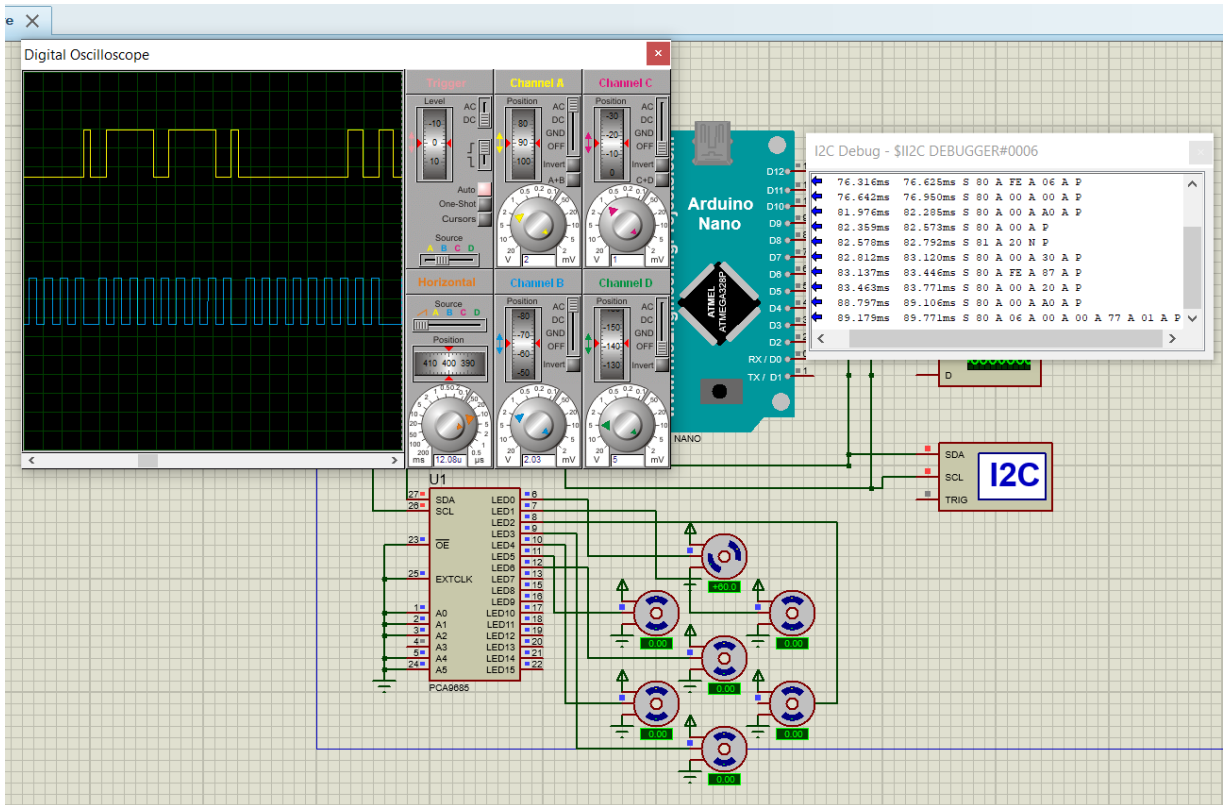


Рисунок 3.13 – Сигнали на лініях SDA та SCL котрі подаються на драйвер PCA9685 з командою підняти перший сервопривід на 60 градусів

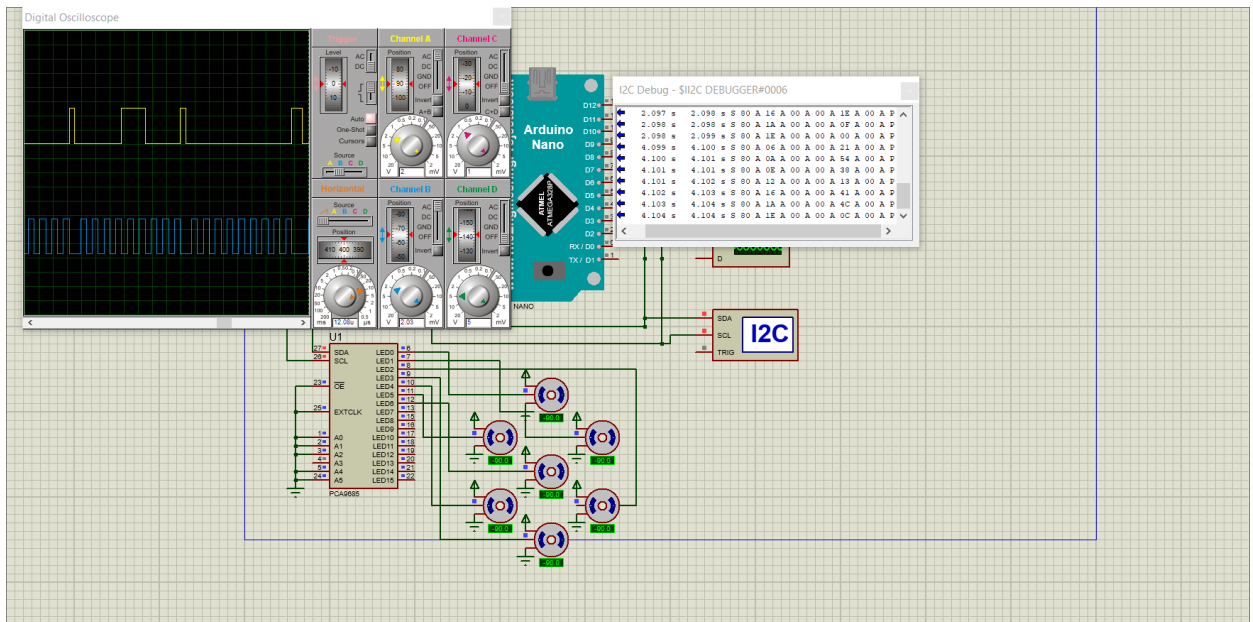


Рисунок 3.14 – Сигнали на лініях SDA та SCL котрі подаються на драйвер PCA9685 з командою повернути всі сервоприводи на 90 градусів

У результаті проведених експериментів були отримані показники швидкості передачі та пропускну здатності нашої шини I2C. Ці дані дозволили нам оцінити продуктивність шини та зробити висновки про її ефективність у використанні нашою радіотехнічною системою.

Крім того, в даному розділі ми також протестували основний лістинг програми, який використовується для керування серводвигунами через драйвер PCA9685. Цей код був успішно виконаний, і ми змогли спостерігати рухи серводвигунів та переконатися в їх правильній роботі, що наведено на рисунку 3.15.

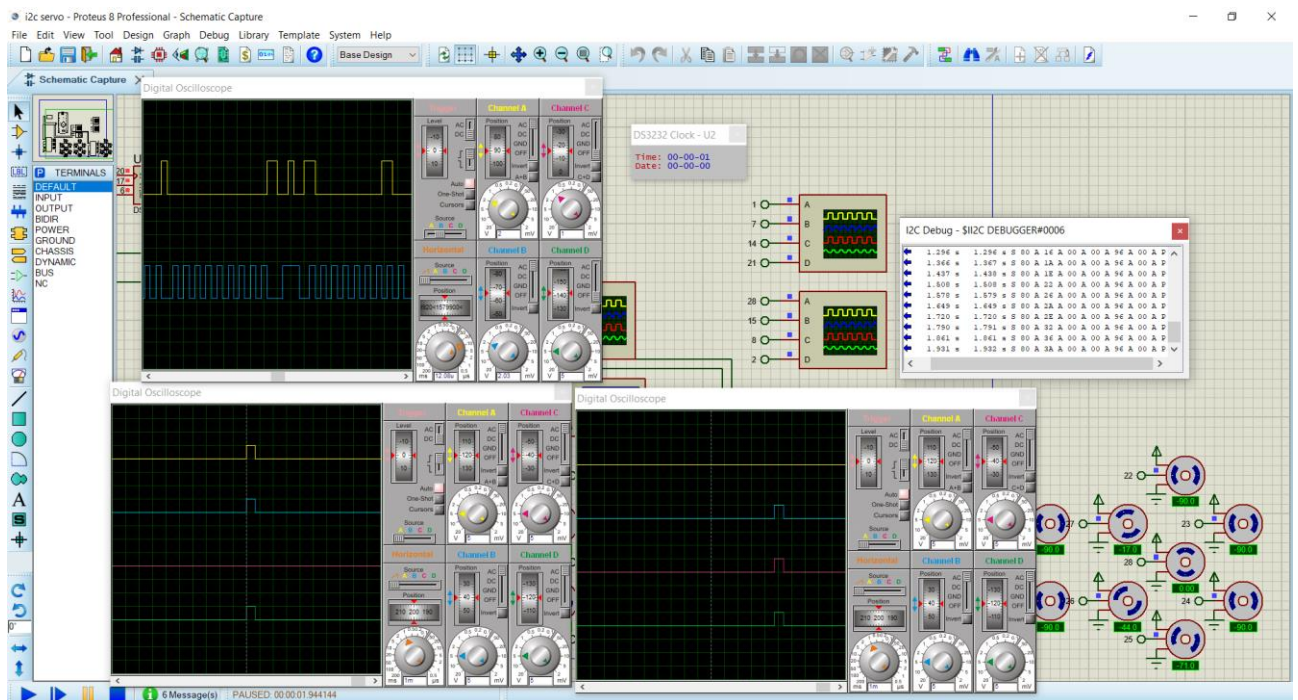


Рисунок 3.15 – Результат роботи проектованої схеми та її лістингу в роботі, зображено осцилограми двох ліній шини I2C та кожного сьомого сервоприводу

В даному підрозділі було представлено детальний опис проведених експериментів, результати тестування та аналіз отриманих даних. Загалом, це дослідження надає цінну інформацію про роботу шини I2C та її використання в радіотехнічній системі відображення інформації.

## 4 ОХОРОНА ПРАЦІ

На сьогоднішній день охорона праці вирішується на національному рівні з метою усунення негативного впливу технологічних процесів на здоров'я людей та поліпшення умов праці. В контексті розробки радіотехнічної системи відображення інформації, яка базується на комп'ютерному проектуванні та моделюванні, основна увага приділяється безпеці праці та покращенню робочих умов.

При проектуванні радіотехнічної системи відображення інформації на сервоприводах враховуються параметри та елементи, що впливають на розробку рішень з охорони праці. Проводиться аналіз потенційно небезпечних та шкідливих факторів, які можуть призвести до травм або пошкодження здоров'я працівників. Застосовується методика аналізу цих факторів та визначення методів управління їх безпекою.

Необхідність вивчення та дослідження охорони праці в рамках радіотехнічної системи відображення інформації полягає у розробці та впровадженні ефективних заходів безпеки, що забезпечать найвищий рівень захисту працівників від можливих небезпек і ризиків, пов'язаних з експлуатацією такої системи.

Ця бакалаврська дипломна робота спрямована на вивчення теоретичних аспектів радіотехнічної системи відображення інформації на сервоприводах, а також на проведення експериментальних досліджень з метою забезпечення безпеки праці під час використання такої системи. Отримані результати можуть бути використані для подальшого вдосконалення та розробки заходів безпеки в подібних радіотехнічних системах.

Таким чином, дослідження охорони праці у контексті радіотехнічної системи відображення інформації на сервоприводах є актуальним завданням, яке потребує ретельного аналізу, проведення експериментальних досліджень та розробки ефективних заходів безпеки. Отримані результати можуть сприяти поліпшенню умов праці та забезпеченню безпеки працівників у галузі радіотехніки.

У робочій зоні можуть виникати фізичні фактори, такі як занадто висока або низька температура повітря, недостатня або надмірна рухливість повітря, забрудненість повітря пилом, недостатня освітленість та відсутність природного освітлення, небезпечний рівень напруги в електричних колах, що може викликати ураження людини, підвищений рівень шуму та вібрації, а також занадто висока або низька вологість повітря.

Психофізіологічні фактори також можуть впливати на працівників. Це можуть бути фізичні перевантаження, спричинені динамічними рухами або фізичними навантаженнями, а також нервово-психічні перевантаження, пов'язані з монотонністю роботи, емоційними стресами та надмірним навантаженням.

Наступні розділи детально розглянуть вищезазначені аспекти охорони праці, пов'язані з об'єктом проектування. Такий підхід допоможе розробити ефективні рішення з охорони праці, забезпечуючи безпеку та здоров'я працівників, а також дотримання відповідних норм та стандартів.

#### 4.1 Технічні рішення з безпечного виконання роботи

##### 4.1.1 Обладнання приміщення та робочого місця

Організація робочого місця та врахування антропометричних, фізичних і психологічних вимог є ключовими аспектами для досягнення оптимального розташування всіх компонентів робочого місця. Крім того, характер роботи також має значний вплив. При організації простору для розробки робочого місця необхідно враховувати наступні основні умови: оптимальне розташування обладнання, яке входить до складу робочого місця, а також наявність достатнього робочого простору, що дозволяє виконувати всі необхідні рухи та переміщення.

1. Для забезпечення чотирьох робочих місць [17] з комп'ютерами необхідне приміщення площею 40 м<sup>2</sup> та стелею висотою 3 метри. У цьому просторі будуть розташовані чотири робочі зони з комп'ютерами, кожна з яких включатиме робочий стіл площею 2 м<sup>2</sup>, стілець та персональний комп'ютер, що

складається з монітора, системного блоку, клавіатури та миші. Робочі місця розробників програмного забезпечення (ПЗ) повинні відповідати наступним вимогам для забезпечення комфорту та ефективності роботи:

2. Ергономіка: Робочі столи повинні мати достатню площу для комфортного розміщення комп'ютерної техніки та документів. Регульовані столи та стільці дозволять працівникам знаходити оптимальну позицію для запобігання неприємним фізичним навантаженням.

3. Освітлення: Необхідно забезпечити достатнє природне або штучне освітлення, щоб уникнути напруження очей та покращити видимість на екрані комп'ютера. Рекомендується використовувати регульоване освітлення для індивідуального налаштування.

4. Вентиляція: Ефективна система вентиляції забезпечить свіже повітря та комфортну температуру у приміщенні, запобігаючи перегріву комп'ютерної техніки та забезпечуючи здорові умови праці.

5. Акустична ізоляція: Для зменшення рівня шуму [18] та відволікань рекомендується застосовувати матеріали звукоізоляції на стінах та підлозі, а також використовувати навушники або інші засоби для зменшення запобігання світлових відблисків:

– Застосування антивідблискових сіток, спеціальних фільтрів, захисних козирків на вікнах для запобігання світлових відблисків від екрана та клавіатури.

– Уникання розташування робочого місця розробника РЕА в напрямку очей від освітлювачів загального призначення або сонячних променів.

6. Конструкція робочого столу:

– Робочий стіл повинен забезпечувати оптимальне розміщення обладнання, враховуючи його розмір та конструктивні особливості (монітор, клавіатура, принтер, ПК тощо), а також документи, що використовуються розробником.

– Конструкція столу повинна відповідати характеру виконуваної роботи розробника РЕА.



Ці вимоги з охорони праці на робочому місці розробника РЕА допоможуть забезпечити комфортні умови праці, запобігти негативним наслідкам для здоров'я та покращити продуктивність роботи.

Площа приміщення, яка припадає на кожного працівника, складає 10 м<sup>2</sup>, об'єм повітря в робочій зоні - 120 куб. м. Знайдені значення відповідають встановленим вимогам, оскільки робоче місце для працівника, який користується комп'ютером, має площу не менше 6 кв. м та об'єм не менше 20 куб. м.

Основними елементами робочого місця розробника є стіл і крісло. Сидяче положення є основним робочим станом, що сприяє зменшенню втоми працівника. Ефективне планування робочого простору передбачає чіткий порядок і постійне розташування предметів, робочих засобів і документів. Предмети, що використовуються найчастіше, повинні бути розміщені в зоні легкого доступу всередині робочого простору.

Для забезпечення комфорту під час роботи необхідно дотримуватися наступних вимог до столу:

- Поверхня столу повинна мати властивості, які уникатимуть відблисків у полі зору, забезпечуючи комфортне сприйняття робочого середовища.

- Вибір висоти столу має забезпечувати вільне сидіння в зручній позі і можливість опиратися на підлокітники за необхідності.

- Стіл повинен мати конструкцію з трьома або більше висувними шухлядками для зручного зберігання документації та канцелярських приладів.

- Нижня частина столу повинна бути спеціально спроектована таким чином, щоб працівник зручно сидів, не відчуваючи дискомфорту через стиснення ніг.

- Рекомендована висота робочої поверхні столу має бути в діапазоні 680-760 мм. Рекомендується, щоб висота поверхні, на яку розміщується клавіатура, б лизько 650 мм.

Крім того, важливо забезпечити достатній об'єм повітря в приміщенні. Згідно з вимогами, об'єм повітря на одного працівника повинен бути не меншим

за 20 куб. м. Це допомагає забезпечити свіже та здорове середовище для працівників.

Для досягнення комфортних умов роботи також рекомендується дотримуватися організаційних принципів. Це включає планування робочого простору з чітким розташуванням предметів та засобів, які використовуються часто. Важливо, щоб ці предмети були легко доступні працівникам всередині робочого простору, щоб уникнути зайвих зусиль та втрати часу на пошук.

Враховуючи всі ці фактори, створення комфортного робочого місця для розробника включає в себе правильне планування простору, вибір відповідного столу та крісла, а також забезпечення відповідного об'єму повітря. Дотримання цих вимог допомагає забезпечити здорове та продуктивне середовище роботи.

#### 4.1.2 Електробезпека приміщення

Конструкція робочого стола повинна бути організована таким чином, щоб враховувати необхідне розташування обладнання, такого як монітор, клавіатура, принтер і ПК, а також документів. Важливо враховувати розмір та особливості кожного з цих елементів, а також враховувати характер роботи, яку виконує працівник.

Згідно з вимогами електробезпеки, при проектуванні та експлуатації систем електропостачання для комп'ютерів необхідно дотримуватись вимог безпеки. Це включає створення окремої групової трипровідної мережі для живлення комп'ютерів, периферійних пристроїв та устаткування для обслуговування. У цій мережі повинні бути фазовий, нульовий робочий та нульовий захисний провідники. Заземлення (занулення) електроприймачів здійснюється за допомогою нульового захисного провідника.

За класифікацією небезпеки [19] електротравматизму, дане робоче приміщення відноситься до категорії "без підвищеної небезпеки". Це означає, що тут відсутні фактори, які можуть створити підвищену або особливу небезпеку.

Для запобігання електротравмам в приміщенні потрібно вживати наступні заходи:

1) Забезпечити ізоляцію струмоведучих елементів електроустаткування відповідно до вимог нормативних документів.

2) Використовувати захисне заземлення з використанням природних заземлювачів.

3) Проводити регулярні інструктажі з питань електробезпеки.

Перед початком роботи необхідно дотриматись таких вимог:

1. Перевірити надійність кріплення обладнання на робочому столі, щоб упевнитись в його стійкості та стабільності під час роботи.

2. Перевірити загальний стан обладнання, проводки електромережі, штепсельних вилок, розеток та заземлення захисного екрана. Упевнитись, що всі елементи є справними та відповідають безпечним стандартам.

3. Налаштувати освітленість робочого місця таким чином, щоб вона була комфортною для зору працівника.

4. Встановити висоту стільця або крісла так, щоб воно було зручним для працівника і сприяло правильній позі тіла під час роботи. Забезпечити можливість налаштування нахилу спинки для забезпечення комфорту.

5. Увімкнути комп'ютерне обладнання в послідовності, що забезпечує безпечне включення. Починати зі стабілізатора або блоку безперервного живлення, далі увімкнути відеомонітор, системний блок та принтер (якщо потрібно).

6. Налаштувати яскравість та контрастність екрану відеомонітора для оптимального комфорту і чіткості зображення.

7. У разі виявлення несправностей у комп'ютерній техніці необхідно повідомити керівництво або відповідний відділ, щоб забезпечити швидке вирішення проблеми.

Виконання цих рекомендацій сприятиме безпеці та комфорту під час роботи з комп'ютером, зменшує ризик електротравм та сприяє ефективності праці.

## 4.2 Технічні рішення з гігієни праці та виробничої санітарії

### 4.2.1 Мікроклімат

Мікроклімат виробничих приміщень регулюється в залежності від теплових характеристик приміщення, рівня складності робіт і сезону. Відповідно до стандарту ДСН 3.3.6.042-99[20], мікроклімат виробничих приміщень визначає умови внутрішнього середовища, які впливають на тепловий комфорт працівників. Ці умови охоплюють температуру, вологість, швидкість руху повітря, температуру поверхонь та рівень опромінення.

Мікроклімат регулюється з урахуванням характеристик приміщення, рівня складності робіт та сезону. Дослідження, що стосуються системи радіотехнічного відображення інформації на сервоприводах, відносяться до категорії 1а. Допустимі параметри [21] мікроклімату для категорії 1а наведені в таблиці 4.1.

Це означає, що виробниче приміщення, де проводяться дослідження, повинно відповідати встановленим нормам щодо температури, вологості, руху повітря та інших параметрів, щоб забезпечити комфортні умови для працівників.

Таблиця 4.1 – Параметри мікроклімату

Період року	Допустимі		
	t, °C	W, %	V, м/с
Теплий	22-28	55	0,1-0,2
Холодний	21-25	75	0,1

За допомогою різних заходів забезпечується відповідний мікроклімат, який відповідає нормативам. Ці заходи включають:

1. Вентиляція приміщення, що забезпечує достатню циркуляцію повітря.
2. Регулювання опалення та кондиціонування повітря, що дозволяє підтримувати оптимальну температуру.

3. Система запобігання задимлення, яка забезпечує видалення диму та забрудненого повітря з приміщення.

4. Утримання оптимальної вологості, що досягається за допомогою контролю та регулювання рівня вологості повітря.

Ці заходи мають на меті забезпечити комфортні умови для працівників і забезпечити дотримання вимог щодо мікроклімату відповідно до нормативів.

#### 4.2.2 Склад повітря робочої зони

У сучасній техніці широко використовуються різноманітні речовини, які можуть потрапляти в атмосферу і становити загрозу здоров'ю людей. Токсичні речовини [22] є тими, що при неправильному використанні або внаслідок порушення технологічних процесів можуть спричинити професійні захворювання, промислові травми або вплинути на загальний стан здоров'я людини.

Токсичні речовини у робочій зоні можуть поширюватися у формі парів, газів або пилу. Вплив цих речовин на організм людини залежить від їх хімічного складу, розміру, форми частинок та їх концентрації. Найнебезпечнішими є фінодисперсний пил (частинки розміром менше 5 мкм) та гострокрайовий пил, які можуть глибоко проникати в легені.

Таблиця 4.2 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м <sup>3</sup>	ГДК, мг/м <sup>3</sup>	Клас небезпечності
	Максимально разова	Середньо добова	
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	1

У приміщенні, де проводяться дослідження властивостей Радіотехнічної системи відображення інформації, може бути підвищена концентрація пилу та

озону. Максимально допустимі рівні шкідливих речовин, які можуть бути присутні у цьому приміщенні (згідно з ДСН 3.3.6.042-99), наведені в таблиці 4.2.

Для забезпечення якості повітря у робочій зоні рекомендується застосовувати механічну вентиляцію та забезпечувати регулярне прибирання та провітрювання.

#### 4.2.3 Виробниче освітлення

Освітлення на робочих місцях має за мету створення необхідних умов для виробничого процесу, безпечного переміщення людей та руху транспорту, і його наявність є обов'язковою в усіх виробничих приміщеннях.

Щодо природного освітлення, рекомендується використовувати бічне освітлення, якщо це можливо, з одного боку.

Згідно з сучасними стандартами, мінімальний рівень освітленості встановлюється з урахуванням зорової активності, контрастності об'єкта та фону.

Один з нормативних параметрів для природного освітлення - коефіцієнт природного освітлення (КПО). Значення КПО встановлюються залежно від виду зорової діяльності. Норми освітленості для штучного освітлення та КПО (для регіону зі світловим кліматом III) при природному та комбінованому освітленні наведені в таблиці 4.4.

Таблиця 4.4 – Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнення	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КПО, $e_n$ , %			
						Штучне освітлення	Комбіноване	Природне освітлення	Сумісне освітлення		
										Загальне	Верхнє або верхнє і бокове
Високої точності	Від 0,15 до 0,3	III	г	великий	світлий	1000	300	7	2,5	4,2	1,5

Для забезпечення відповідного освітлення передбачаються певні заходи, серед яких важливі:

1. Встановлення достатньої кількості освітлювальних приладів, що забезпечує належний рівень освітленості.
2. Використання підсвічування, яке може бути корисним для покращення видимості на певних об'єктах або ділянках.
3. Розміщення джерел світла на оптимальній висоті, що забезпечує рівномірне розподіл світла та уникнення непотрібних тіней.
4. Регулярне обслуговування освітлювальної системи для підтримання її ефективності та безпеки.

Ці заходи сприяють створенню належних умов освітлення, що важливо для забезпечення продуктивності, безпеки та комфорту працівників.

#### 4.2.4 Виробничий шум

Під час проведення досліджень властивостей Радіотехнічної системи відображення інформації на сервоприводах, на робочому місці дослідника виникає небажаний шум, що походить від функціонування різноманітних системних блоків, принтерів обладнання для кондиціонування повітря, вентиляторів систем охолодження та сервоприводів.

Нормативні вимоги щодо шуму, ультразвуку та інфразвуку на робочому місці розглянуті в ДСН 3.3.6.037-99 [23]. Залежно від характеру робіт та особливостей шуму, необхідно дотримуватися припустимих рівнів звукового тиску, які відповідають гігієнічним нормам. Рівень звуку  $L_A$  не повинен перевищувати 50 дБА, що можна знайти в таблиці 4.5.

Забезпечення відповідної акустичної комфортності на робочому місці є важливим для збереження здоров'я та забезпечення ефективної роботи дослідника.

Для забезпечення задовільного рівня шуму в приміщенні рекомендується використовувати комп'ютери, які не виробляють шуму, і мають пасивну систему охолодження.

Таблиця 4.5 – Допустимі рівні звукового тиску та звуку

Характер робіт	Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах зі середньгеометричними частинами (Гц)									Допустимий рівень звуку, дБА
	32	63	125	250	500	1000	2000	4000	8000	
Конструювання та проектування	86	71	61	54	49	45	42	40	38	50



#### 4.2.5 Виробничі випромінювання

У проекті присутні комп'ютери, які генерують електромагнітне випромінювання. Для забезпечення безпеки працівників, необхідно дотримуватися встановлених норм та вимог щодо рівнів електромагнітного випромінювання та магнітних полів. Важливо забезпечувати, щоб рівні інфрачервоного випромінювання залишалися в межах припустимих лімітів. Також необхідно дотримуватися вимог щодо рівнів ультрафіолетового випромінювання відповідно до нормативів. Максимально допустима напруженість електростатичного поля на робочих місцях повинна знаходитися в межах встановлених рівнів наведених в таблиці 4.6.

Таблиця 4.6 – Допустимі рівні випромінювань моніторів ПК

Вид поля	ТСО	MPR II
Змінне електричне поле 5 Гц – 2 кГц 2 кГц – 400 кГц	10 В/м 1 В/м на відстані 0,3 м від центра екрана і 0,5 м навколо монітора	2,5 В/м 2,5 В/м на відстані 0,5 м навколо монітора
Змінне магнітне поле 5 Гц – 2 кГц 2 кГц – 400 кГц	250 нТл 200 мА/м 25 нТл 20 мА/м на відстані 0,3 м від центра екрана і 0,5 м навколо монітора	250 нТл 200 мА/м 25 нТл 20 мА/м на відстані 0,5 м навколо монітора

Для забезпечення захисту [24] людини від електромагнітного випромінювання, застосовуються різні заходи, такі як обмеження часу перебування працівника в зоні випромінювання та встановлення раціональних режимів експлуатації комп'ютерів і роботи працівника.

З метою забезпечення безпеки працівників, рекомендується обмежувати тривалість перебування працівника в зоні, де випромінюється електромагнітне випромінювання. Це можна досягти шляхом розподілу робочого часу та певними перервами, що дозволять працівникам віддалитися від джерел випромінювання на певний період часу.

Крім того, важливо встановити раціональні режими експлуатації комп'ютерів та організувати роботу працівника відповідно до норм безпеки. Це включає правильне налаштування параметрів комп'ютерів, зокрема частоти оновлення монітора та інших параметрів, які можуть впливати на рівень електромагнітного випромінювання. Додатково, слід враховувати режими роботи працівника, забезпечуючи достатні перерви для відновлення та відпочинку.

Ці заходи допоможуть зменшити вплив електромагнітних випромінювань на працівників і забезпечити їхню безпеку під час роботи з комп'ютерами.

### 4.3 Пожежна безпека

Запобігання пожежам є пріоритетним завданням, і для цього необхідно усунути можливість утворення горючих або вибухонебезпечних середовищ та джерел запалювання.

На підприємстві необхідно прийняти ефективні заходи з пожежної безпеки, які допоможуть знизити ризики для людей і обмежити матеріальні збитки у разі пожежі. Забезпечення пожежної безпеки об'єкта включає в себе систему запобігання пожежам, систему протипожежного захисту і організаційно-технічні заходи.

Основна мета пожежної безпеки [25] полягає в тому, щоб відповідати нормативним вимогам та запобігати виникненню пожеж. У разі виникнення пожежі, метою є обмеження поширення вогню, вчасне виявлення та загашення пожежі, а також захист людей і матеріальних цінностей. Зазначене приміщення класифікується як категорія "Д".

Варто відмітити, що пожежна безпека є важливою складовою безпеки на робочому місці, і вона повинна бути серйозною перевіреною та врахована в усіх аспектах діяльності підприємства.

#### 4.3.1 Технічні рішення системи запобігання пожежі

У приміщенні існують фактори, які можуть спричинити пожежу. Серед них:

- Перевантаження електромережі та перегрів струмоведучих частин і з'єднань.
- Порушення правил експлуатації електротехнічного обладнання.
- Недостатня електрична безпека, така як перевантаження електричних мереж і недостатнє заземлення, що може спричинити перегрів і коротке замикання.
- Порушення правил використання електроприладів, наприклад, неправильне підключення або залишення увімкненими приладів без нагляду.
- Несправність електричного обладнання, включаючи пошкоджену ізоляцію, старі та зношені кабелі, дефекти в електричних приладах.
- Система запобігання пожежі передбачає наступні заходи:
- Регулярна перевірка цілісності ізоляції.
- Наявність спеціально відведених місць для куріння.
- Проведення періодичних навчань та інструктажів з протипожежної безпеки.
- Уникання накопичення горючих матеріалів у приміщенні.
- Встановлення системи захисту від атмосферних розрядів.

Ці заходи мають на меті зменшити ризик пожежі, зберегти безпеку людей і майна, а також забезпечити відповідність нормам протипожежної безпеки.

#### 4.3.2 Технічні рішення системи протипожежного захисту

В приміщенні можна використовувати автоматичну систему пожежогасіння [26], яка буде ефективно реагувати на випадок виникнення пожежі. Для цього рекомендується встановити установку порошкового пожежогасіння, що використовує нетоксичний порошок як вогнегасну речовину. Це дозволить ефективно загасити пожежу, забезпечуючи безпеку приміщення.

Крім того, слід встановити пожежну сигналізацію, яка буде

відслідковувати наявність пожежі в приміщенні та надсилати відповідні сигнали. Це допоможе швидко виявити загоряння та розпочати негайні заходи по гасінню пожежі.

Після завершення роботи всі електроприлади, а також їх живлення повинні бути вимкнені, за винятком протипожежних та охоронних систем. Це зменшить ризик виникнення пожежі внаслідок перегріву та короткого замикання електрообладнання.

Окрім цього, електропроводи для підключення комп'ютерів та інших приладів мають бути прокладені по негорючих конструктивних елементах. Це забезпечить додатковий рівень безпеки, оскільки негорючі матеріали мають меншу ймовірність сприяти розповсюдженню пожежі.

Відповідно до встановлених норм, на кожні 20 м<sup>2</sup> площі приміщення з відповідною пожежовибухонебезпекою та можливими класами пожеж (А, В, і Е), рекомендується розміщувати один порошковий або вуглекислотний вогнегасник масою заряду від 3 до 5 кг. Також, на кожному поверсі, де знаходиться приміщення, слід забезпечити на несправності пожежної техніки слід повідомляти пожежній охороні. Ці заходи сприятимуть зменшенню ризику виникнення пожежі та забезпечать безпеку на робочому місці. В приміщенні також буде розташовуватися один порошковий вогнегасник ПВ-3, який буде розміщуватись на висоті не більше 1,5 м від рівня підлоги до нижнього торця вогнегасника, знаходитися на відстані від дверей, достатній для їх повного відчинення. Для позначення місцезнаходження вогнегасника буде встановлений вказівний знак, який розташовують на видних місцях на висоті 2,0 - 2,5 м від рівня підлоги.

## ВИСНОВКИ

В бакалаврській дипломній роботі було проведено аналіз радіотехнічної системи відображення інформації на сервоприводах з використанням семисегментної індикації та модуля Arduino Nano. Було розглянуто актуальність та потребу відображення інформації, а також визначено типи даних, переваги та можливості такої системи.

Аналіз семисегментної індикації показав, що вона є ефективним методом відображення інформації, забезпечуючи високу читабельність та можливість відображення різних типів даних. Використання сервоприводів дозволяє досягти точного позиціонування та керування кутом обертання рухомих частин системи. Шина I2C є зручним засобом взаємодії між модулями системи, а модуль Arduino Nano виконує роль керуючого пристрою, забезпечуючи взаємозв'язок між компонентами системи.

Для досягнення мети роботи були сформульовані задачі, включаючи аналіз семисегментної індикації, вивчення параметрів сервоприводів, розробку алгоритмів керування, дослідження можливостей шини I2C та встановлення взаємозв'язку між компонентами системи.

Результати аналізу та досліджень показали, що радіотехнічна система відображення інформації на сервоприводах є актуальною і має численні переваги. Вона може знайти застосування в різних галузях, включаючи автоматизацію, вбудовані системи та інші області, де точна та зручна відображення інформації є важливим фактором.

У рамках дослідження було розроблено програмне забезпечення для радіотехнічної системи відображення інформації на сервоприводах. Також було проведено комп'ютерне моделювання окремих блоків системи, зокрема сервоприводів та шини I2C, у моделюючому пакеті Proteus 8. Це дозволило оцінити швидкість передачі й пропускну здатність шини I2C з візуалізацією цього процесу за допомогою осцилографа та терміналу I2C.

Розроблена радіотехнічна система відображення інформації на сервоприводах на основі семисегментної індикації та модуля Arduino Nano має

потенціал для широкого застосування. Її переваги полягають у точному відображенні різних типів даних, гнучкості налаштування, можливості взаємодії з іншими пристроями за допомогою шини I2C, а також простоті інтеграції з рухомими частинами системи.

Завдяки проведеному аналізу, впровадженню відповідних рішень та подальшій розробці, ця радіотехнічна система може стати цінним інструментом для покращення відображення інформації на сервоприводах, що відкриває широкі перспективи для подальших досліджень та розробок у цій галузі. Продовження роботи включатиме розгляд реалізації системи відображення інформації на основі проведеного аналізу, детальне розгляд алгоритмів керування, взаємодію з модулем Arduino Nano та сервоприводами через шину I2C, а також розробку та впровадження необхідних модулів системи.

Також було розглянуто розділ охорони праці на робочому місці, опрацьовані питання з регуляції мікроклімату та пожежну безпеку й обрано запобіжні заходи та алгоритми дій для забезпечення пожежної безпеки

Отже, бакалаврська дипломна робота присвячена розробці та аналізу радіотехнічної системи відображення інформації на сервоприводах з використанням семисегментної індикації та модуля Arduino Nano. В результаті аналізу було виявлено, що ця система є актуальною та має значні переваги, зокрема високу читабельність і можливість відображення різних типів даних. Розроблене програмне забезпечення та проведене комп'ютерне моделювання підтвердили функціональні можливості системи та її потенціал для застосування у різних галузях, включаючи автоматизацію, вбудовані системи та інші області. Робота над цією системою може бути продовжена з метою розробки детальних алгоритмів керування, взаємодії з іншими компонентами системи та впровадження додаткових модулів для поліпшення її функціональності.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Андерсон, Л., та ін. (2018). "Proteus: комплексний засіб моделювання вбудованих систем." Матеріали міжнародної конференції з вбудованих систем, с. 45-58.
- 2 Андерсон, Дж. (2017). "Моделювання електронних систем за допомогою Proteus." Видавництво Elsevier.
- 3 Джонсон, М. (2017). "Застосування семисегментної індикації в інформаційній візуалізації." Журнал людино-комп'ютерної взаємодії, Том 42, № 2, с. 221-235.
- 4 Джонсон, А. (2019). "Технології відображення: вступ." Видавництво Оксфордського університету.
- 5 Специфікація шини I2C. URL: <https://www.i2c-bus.org/specification/>
- 6 Документація Arduino URL: <https://www.arduino.cc/en/Guide/ArduinoNano>
- 7 Редді, Р. Аджай Кумар, Кришна, В. П. Венката, Редді, Р. П. Сурендер. (2018). "Проектування та розробка системи відображення в реальному часі на основі годинника за допомогою Arduino". Міжнародний журнал інженерії та технології, 7(4.19), 127-131.
- 8 Документація на модуль RTC DS3231 URL: <https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html>
- 9 Сервопривід SG-90 URL: <https://proflin.in.ua/ua/elektronika/servoprivod-sg90>
- 10 Документація PCA9685 URL: <https://www.nxp.com/products/analog/interfaces/ic-bus/ic-led-controllers/16-channel-12-bit-pwm-fm-plus-ic-bus-led-controller:PCA9685>
- 11 Браун, А., та Міллер, К. (2019). "Вступ до програмування та взаємодії з Arduino." Журнал IEEE з питань освіти, Том 62, № 3, с. 180-195.
- 12 Браун, Т. (2018). "Програмування Arduino: крок за кроком." Видавництво Packt.
- 13 Сміт, Дж. (2018). "Розширені техніки керування сервоприводами." Міжнародний журнал електроніки та зв'язку, Том 73, с. 124-138.

- 14 Сміт, Р. (2017). "Основи систем сервоприводів." Видавництво McGraw-Hill.
- 15 Документація Proteus 8 URL: <https://www.labcenter.com/>
- 16 Томпсон, Р., та Вілсон, С. (2020). "Шина I2C: комплексне дослідження." Журнал електротехніки, Том 89, № 4, с. 301-315.
- 17 Правила улаштування електроустановок URL: <http://www.energiy.com.ua/PUE.html>
- 18 ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. URL: <http://document.ua/sanitarni-normi-virobnichogo-shumu-ultrazvuku-ta-infrazvuku-nor4878.html>
- 19 ДСТУБ В.2.5-82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. К. : ДП «УкрНДНЦ», 2016. 109 с
- 20 ДСНіПЗ.3.6.096-2002. Державні санітарні норми і правила при роботі з джерелами електромагнітних полів. URL: <http://zakon2.rada.gov.ua/laws/show/z0203-03>
- 21 ДСТУ ОHSAS 18002:2015. Системи управління гігієною та безпекою праці. Основні принципи виконання вимог ОHSAS 18001:2007 (OHSAS 18002:2008, IDT). К. : ГП «УкрНИУЦ», 2016. 21 с.
- 22 Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - - URL:: <http://zakon4.rada.gov.ua/laws/show/v4137400-86>
- 23 ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. – URL: <http://mozdocs.kiev.ua/view.php?id=1972>
- 24 НПАОП 40.1-1.32-01 Правила будови електроустановок. Електрообладнання спеціальних установок URL: <https://zakon.rada.gov.ua/rada/show/v0272203-01#Text>
- 25 ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги. URL: [http://www.poliplast.ua/doc/dbn\\_v.1.1-7-2002.pdf](http://www.poliplast.ua/doc/dbn_v.1.1-7-2002.pdf)
- 26 Наказ Міністерства внутрішніх справ України «Про затвердження Правил експлуатації та типових норм належності вогнегасників». URL: <https://zakon.rada.gov.ua/laws/show/z0225-18#Text>.



Додаток А  
(обов'язковий)

**ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (БАКАЛАВРСЬКОЇ)  
ДИПЛОМНОЇ РОБОТИ**

**РАДІОТЕХНІЧНА СИСТЕМА ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА  
СЕРВОПРИВОДАХ**

ПРОТОКОЛ  
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Радіотехнічна система відображення інформації на сервоприводах»

Тип роботи: БДР  
(БДР, МКР)


Підрозділ ІРТС, ІЕС  
(кафедра, факультет)

**Показники звіту подібності Unicheck**

Оригінальність 89,3% Схожість 10,7%

Аналіз звіту подібності (відмітити потрібне):

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

Особа, відповідальна за перевірку   
(підпис)

Олександр ЗВЯГІН  
(прізвище, ініціали)

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

Автор роботи   
(підпис)

Роман ПЕХОТА  
(прізвище, ініціали)

Керівник роботи   
(підпис)

Максим ПРИГУЛА  
(прізвище, ініціали)

Додаток Б  
(обов'язковий)

**ІЛЮСТРАТИВНА ЧАСТИНА**

**РАДІОТЕХНІЧНА СИСТЕМА ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ  
НА СЕРВОПРИВОДАХ**

Виконав: студент 2(4)-го курсу, групи ТКР-21мс  
спеціальності 172 – Телекомунікації та радіотехніка  
(шифр і назва напрямку підготовки, спеціальності)

Пехота Р.В.

(прізвище та ініціали)

Керівник: к.т.н., ст.викл. ІРТС

Притула М.О.

(прізвище та ініціали)

« 15 » 06 2023 р.

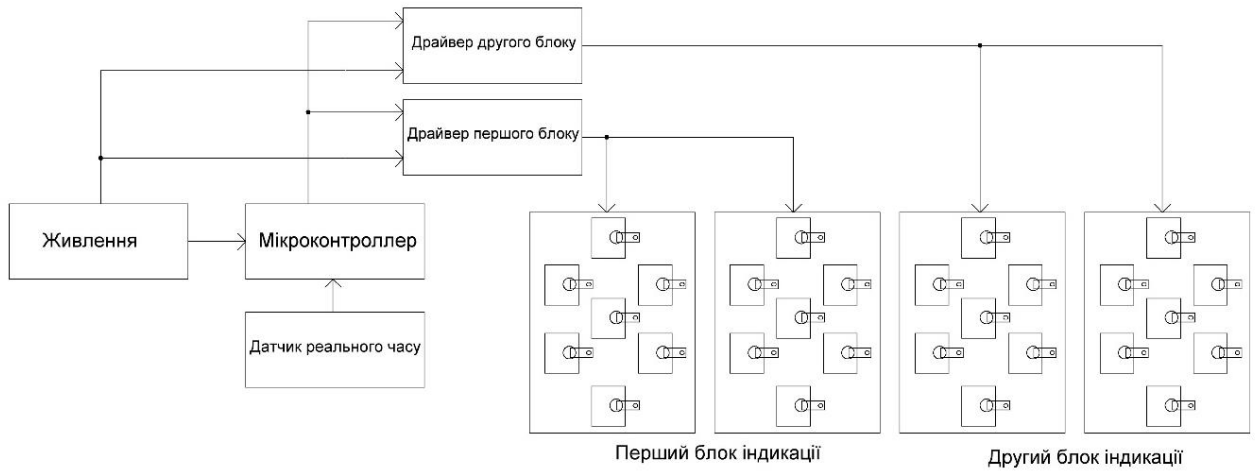


Рисунок 1 – Структурна схема радіотехнічної системи відображення інформації на сервоприводах

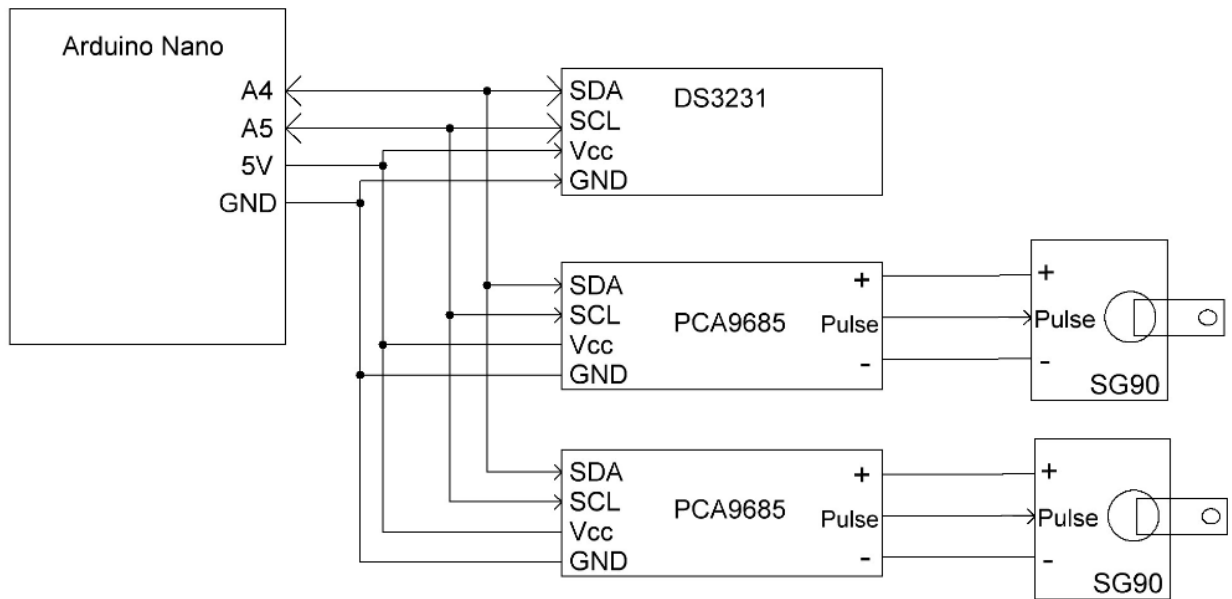


Рисунок 2 – Функціональна схема радіотехнічної системи відображення інформації на сервоприводах

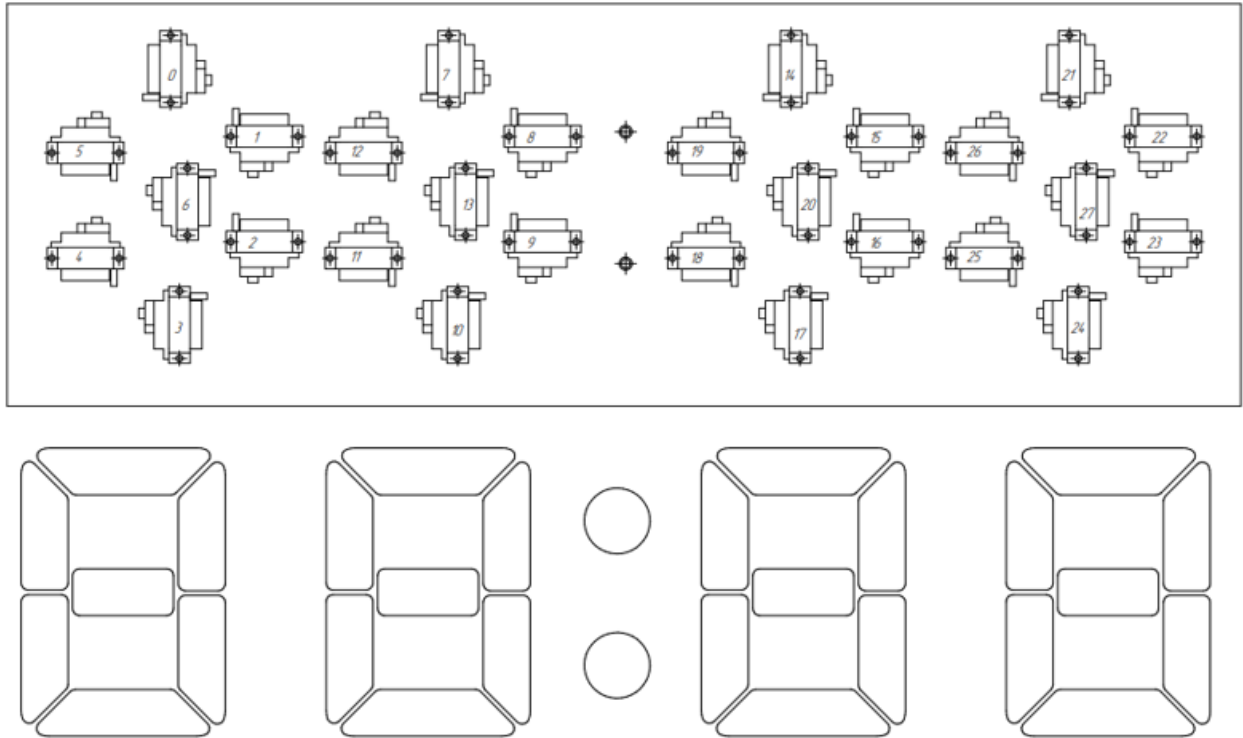


Рисунок 3 – Схема індикації Радіотехнічної системи відображення інформації

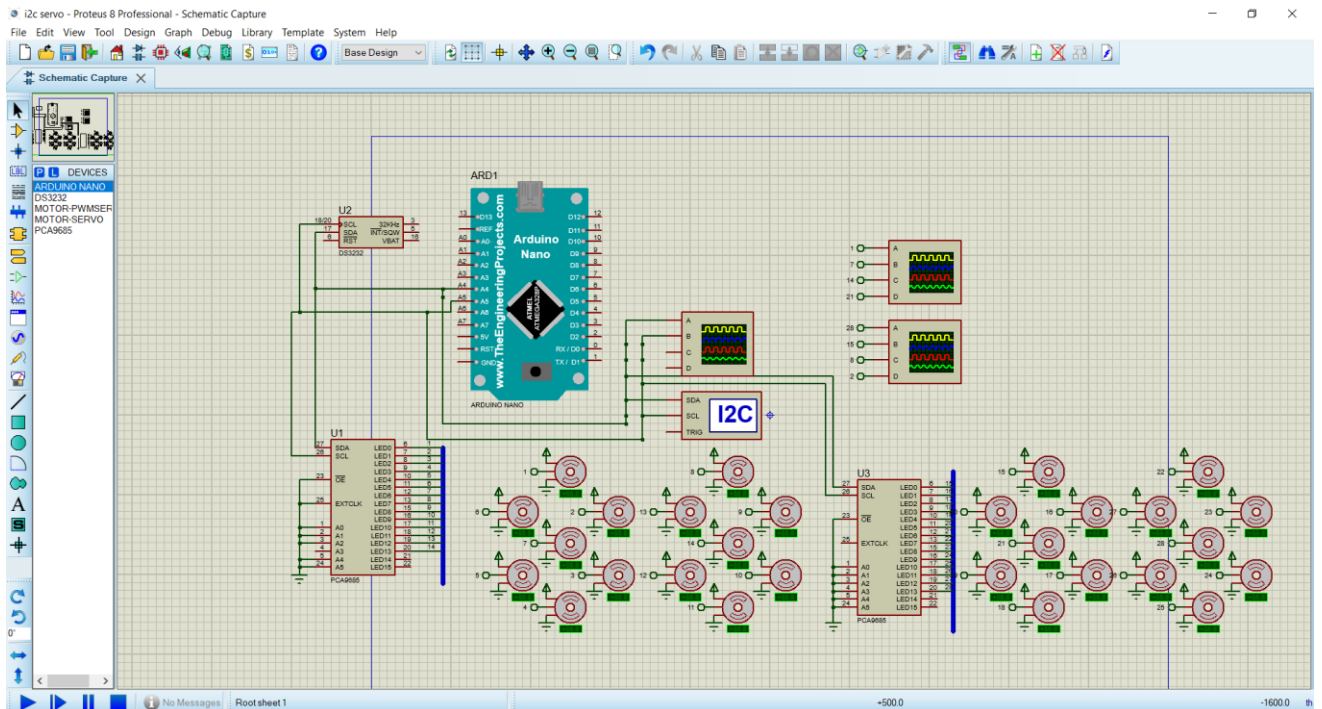


Рисунок 4 – Електронний макет для експериментальних досліджень

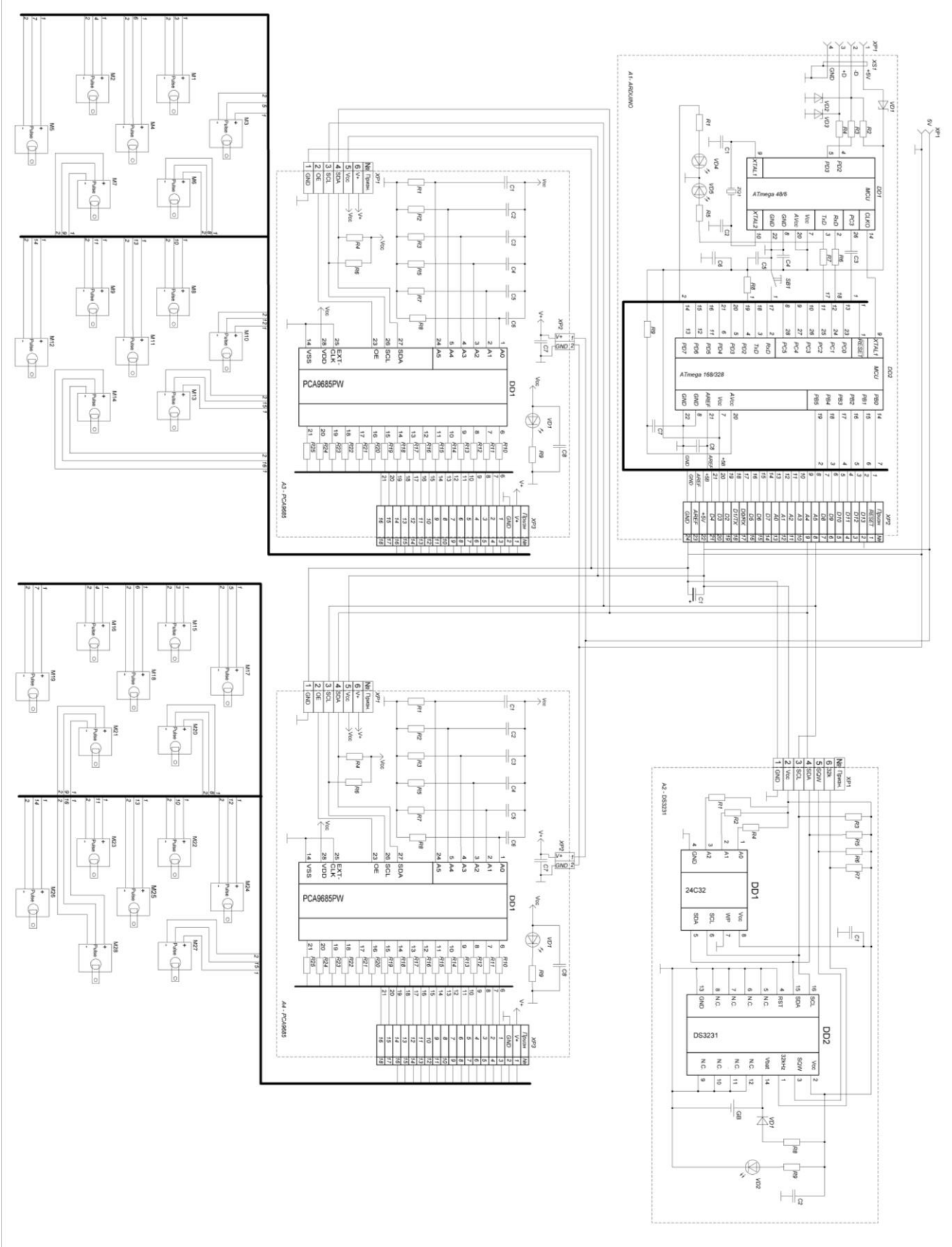


Рисунок 5 – Схема електрична принципова Радіотехнічної системи відображення інформації на сервоприводах

Поз. познач.	Найменування	Кіл.	Примітка					
A1	Платформа Arduino Nano	1						
	<u>Конденсатори</u>							
C1, C2	SMD-0805- 0,1 мкФ ±10%	2						
C3	SMD-0805- 0,01 мкФ ±10%	1						
C4	SMD-0805- 0,1 мкФ ±10%	1						
C5	SMD-0805- 0,01 мкФ ±10%	1						
C6	SMD-0805- 0,1 мкФ ±10%	1						
C7	SMD-0805- 22 нФ ±10%	1						
C8	SMD-0805- 0,1 мкФ ±10%	1						
	<u>Мікросхеми</u>							
DD1	ATmega8	1						
DD2	ATmega328	1						
	<u>Резистори</u>							
R1	SMD-0805- 300 Ом ±5%	1						
R2, R3	SMD-0805- 1 кОм ±5%	2						
R4	SMD-0805- 10 кОм ±5%	1						
R5, R6	SMD-0805- 300 Ом ±5%	2						
R7, R8	SMD-0805- 300 Ом ±5%	2						
R9	SMD-0805- 10 кОм ±5%	1						
SB1	Кнопка тактова 3x6x2.5 мм SMD (10365)	1						
	<u>Діоди</u>							
VD1	Діод Шоткі SMD (SOT23)BAS54	1						
VD2, VD3	Стабілітрон 3.6В SMD (SOT23)	2						
VD4, VD5	Світлодіод SMD (0805)	2						
08-36.БДР.010.00.000								
Змн.	Арк.	№ докум.	Підпис	Дата	Радіотехнічна система відображення інформації на сервоприводах Перелік елементів	Літ.	Арк.	Аркушів
Розроб.		Пехота Р.В.		12.06.25				
Керівник		Притула М.О.		12.06.25			66	3
Реценз.		Михалевський Д.В.		13.06.25				
Н. Коитр.		Червак О.П.		13.06.25				
Затверд.		Осадчук О.В.		15.06.25				
						Гр. ТКР-21мс		







## Додаток В (довідниковий)

Лістинг програми для Радіотехнічної системи відображення інформації на сервоприводах

```

// ***** НАЛАШТУВАННЯ *****
# define SERVOMIN 150 // мінімальний сигнал серво
# define SERVOMAX 550 // максимальний сигнал серво
# define IR_PIN 2 // пін ІЧ датчика
# define IR_SENSOR 1 // використовувати ІЧ датчик руху
# define IR_TIMEOUT 3000000000000000000000 // таймот датчика
руху ( замовч . 300000мс це 5 хв )
# define STOCK_UP 270 // значення максимального кута (сегмент
піднятий ) за умовчанням
# define STRING_SPEED 500 // швидкість біжучого рядка
// текст рядка , що біжить , налаштовується приблизно у рядку
№100

// ***** БІБЛІОТЕКИ *****
#include <Wire.h>
# include "Adafruit_PWMServoDriver.h"
# include "RTCLib.h"
# include "EEPROM.h"

// ***** ЗМІННИ *****
Adafruit_PWMServoDriver hrs_drv = Adafruit_PWMServoDriver
(0x40);
Adafruit_PWMServoDriver mins_drv = Adafruit_PWMServoDriver
(0x41);

RTC_DS3231 rtc ;

uint16_t servoInterval = 40; //(15) тимчасовий інтервал
прирошення кута для плавного повороту .
// Взято 15мс, тому що при повороті ВСІХ серво цей блок
виконується 18 мс
byte servoStep = 5; //(15) збільшення кута повороту серво .
Зберігається в EEPROM ( servo speed )
uint32_t servoTimer , timeTimer , stringTimer , IRTimer ;

boolean turnFlag ; // прапор на опитування блоку з серво (
розвантажує процесор )
uint16_t angle [28]; // особистий кут кожною серво
uint16_t new_angle [28]; // новий кут . Змінюємо значення масиву
- серво повертається
uint16_t up_angle [28]; // Кут , у якому індикатор піднятий .
Зберігається в EEPROM
boolean clock_state ;
boolean adj_flag , adj_flag2;

```

```

byte adj_servo ;
byte stringCounter ;
boolean stringFlag ;
boolean IRFlag ;

/***** БУКВИ I СИМВОЛИ *****/
// http://www.uize.com/examples/seven-segment-display.html
#define _A 0x77
#define _B 0x7f
#define _C 0x39
#define _D 0x3f
#define _E 0x79
#define _F 0x71
#define _G 0x3d
#define _H 0x76
#define _J 0x1e
#define _L 0x38
#define _N 0x37
#define _O 0x3f
#define _P 0x73
#define _S 0x6d
#define _U 0x3e
#define _Y 0x6e
#define _a 0x5f
#define _b 0x7c
#define _c 0x58
#define _d 0x5e
#define _e 0x7b
#define _f 0x71
#define _h 0x74
#define _i 0x10
#define _j 0x0e
#define _l 0x06
#define _n 0x54
#define _o 0x5c
#define _q 0x67
#define _r 0x50
#define _t 0x78
#define _u 0x1c
#define _y 0x6e
#define _dash 0x40
#define _under 0x08
#define _equal 0x48
#define _empty 0x00

#define _0 0x3f
#define _1 0x06
#define _2 0x5b
#define _3 0x4f
#define _4 0x66
#define _5 0x6d
#define _6 0x7d
#define _7 0x07
#define _8 0x7f

```

```

#define _9 0x6f

byte HEXnumbers [10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d,
0x7d, 0x07, 0x7f, 0x6f};
// текст рядка , що біжить
/*
    byte runningString [] = {_empty , _empty , _empty , _empty
,
    _H, _E, _L, _L, _O, _empty , _empty ,
    _G, _Y, _U, _E, _r, _empty , _empty ,
    _S, _e, _r, _u, _o, _empty , _empty ,
    _C, _L, _O, _C, _H,
    _empty , _empty , _empty , _empty ,
};
*/
byte runningString[] = {_empty , _empty , _empty , _empty ,
    _0, _1, _2, _3, _4, _5, _6, _7, _8, _9,
    _empty , _empty , _empty , _empty ,
};
/***** БУКВИ І СИМВОЛИ *****/

void setup () {
    Serial.begin (9600);
    Serial.setTimeout (300);
    if (!IR_SENSOR) IRFlag = true ; // якщо не використовуємо ІЧ,
дозволити роботу

    // Перевірка RTC
    if ( rtc.begin () ) {
        Serial.println ("RTC OK");
        clock_state = true ;
    } else {
        Serial.println ("RTC ERROR");
        clock_state = false ;
    }
    help (); // Виведення текстового меню
    // старт і налаштування серво шилда
    hrs_drv.begin ();
    hrs_drv.setPWMFreq (60); // Analog servos run at ~60 Hz
updates
    mins_drv.begin ();
    mins_drv.setPWMFreq (60); // Analog servos run at ~60 Hz
updates
    // перший запуск
    if ( EEPROM.read (100) != 10) { // зберігаємо якесь число як
індикатор запуску
        Serial.println (" first start ");
        EEPROM.writeByte (100, 10);
        for ( byte i = 0; i < 28; i++) {
            EEPROM.writeInt (i * 2, STOCK_UP);
            EEPROM.writeByte (90, 15);
        }
    }
}

```

```

servoStep = EEPROM.read (90); // Згадуємо швидкість повороту
// при запуску все серво опустити
for ( int i = 0; i < 28; i++) {
  new_angle [i] = SERVOMIN;
  angle [i] = SERVOMIN;
  up_angle [i] = EEPROM.readInt (i * 2); // відновити з
пам'яті верхні значення індикаторів
  if (i < 14)
    hrs_drv.setPWM (i, 0, angle [i]); // перший шилд

  else {
    mins_drv.setPWM (i - 14, 0, angle [i - 14]); // другий
шилд
  }

  delay (120); //(70) затримка між поворотами , ніж брати
великий струм
}
delay (700);
for ( int i = 0; i < 28; i++) {
// відключити серво
  if (i < 14)
    hrs_drv.setPWM (i, 0, 4096);
  else {
    mins_drv.setPWM (i - 14, 0, 4096);
  }
}
delay (700);
Serial.println (F(" Servo ready "));
timeTimer = 60000; // Штучно переповнити таймер часу _
// щоб годинник відразу показали час (див. timeTick ())
}

void help () {
  Serial.println(F("*****
*****"));
  Serial.println (F("< set-time HH:MM> - set time ( example :
set-time 01:20)"));
  Serial.println (F("< get-time > - return RTC time "));
  Serial.println (F("< start-clock> - start counting time "));
  Serial.println (F("< stop-clock > - stop counting time "));
  Serial.println (F("< start-string > - start running string
"));
  Serial.println (F("< stop-string > - stop running string "));
  Serial.println (F("< all-down > - turn all segments down "));
  Serial.println (F("< all-up > - turn all segments up "));
  Serial.println (F("< down NUMBER> - turn segment NUMBER down
( example : down 10)"));
  Serial.println (F("< up NUMBER> - turn segment NUMBER up (
example : up 10)"));
  Serial.println (F("< speed VALUE> - set servo Speed to VALUE,
default 15 ( example : speed 20)"));
  Serial.println (F("< adj NUMBER VALUE> - set max VALUE for
servo with NUMBER and store it in EEPROM ( example : adj 05300)"));
}

```

```

    Serial.println (F("< help > - print this instruction again
"));
    Serial.println(F("*****
*****"));
}

void loop () {
    serialTick (); // обробка команд з порту
    servoTick (); // плавне управління серво
    timeTick (); // таймер часу ( прапором clock_state )
    displayString (); // відобразити рядок , що біжить (по прапору
stringFlag )
    IRTick (); // Перевірка ІЧ датчика
}
void IRTick () {
    IRFlag = true ;
    // if (IR_SENSOR) {
    // if ( digitalRead (IR_PIN)) { // якщо спрацював датчик руху
    // IRTimer = millis (); // скинути таймер
    // if (! IRFlag ) IRFlag = true ; // увімкнути годинник
    // } else { // якщо датчик показує 0
    // if ( millis () - IRTimer > IR_TIMEOUT) { // перевірка на
тайм
    // if ( IRFlag ) IRFlag = true ; //( false ) // відключити
ГОДИННИК
    // }
    // }
    // }
}
void displayString () {
    if ( stringFlag ) {
        if ( millis () - stringTimer > STRING_SPEED) { // Таймер
швидкості рядка
            stringTimer = millis ();
            byte servoNum = 0; // з 0 серви
            for ( byte i = 0; i < 4; i++) { // для 4 індикаторів
                for ( byte j = 0; j < 7; j++) { // кожен сегмент
                    boolean thisBit ;
                    thisBit = runningString [ stringCounter + i] & (1 <<
j); // отримати 0 або 1
                    if ( thisBit ) new_angle [ servoNum ] = up_angle [
servoNum ]; // підняти
                    else new_angle [ servoNum ] = SERVOMIN; // опустити
                    servoNum ++; // Перейти до наступної серво
                }
            }
            stringCounter ++;
            if ( stringCounter > ( sizeof ( runningString ) - 4))
stringCounter = 0;
            turnFlag = true ;
        }
    }
}
void timeTick () {

```

```

    if ( clock_state ) {
        if ( millis () - timeTimer > 60000) { // відраховуємо
хвилину
            timeTimer = millis ();
            // кожна хвилину беремо час з RTC тип

            DateTime now = rtc.now ();
            Serial.print ( " New minute . Time is " );
            Serial.print ( now.hour ()); Serial.print ( ":" );
Serial.println ( now.minute ());

            // забиваємо масив для відображення
uint8_t indicators [4];
            indicators [0] = now.hour () / 10; // десятки годин
            indicators [1] = now.hour () % 10; // одиниці годин
            indicators [2] = now.minute ()/10; // десятки хвилин
            indicators [3] = now.minute () % 10; // одиниці хвилин
            turnFlag = true ; // дозволити поворот серво
            byte servoNum = 0;
            for ( byte i = 0; i < 4; i++) { // для 4 індикаторів
                for ( byte j = 0; j < 7; j++) { // кожен сегмент
                    boolean thisBit ;
                // thisBit приймає 0 або 1 - підняти чи опустити сегмент.
                // HEXnumbers розбивається на біти , наприклад 0000110 це цифра
1
                // і побитно приймається рішення , підняти або опустити сегмент
                    thisBit = HEXnumbers [ indicators [i]] & (1 << j);
                    if (i == 0 && indicators [0] == 0) thisBit = 0; // не
показувати нуль у десятках годин
                    if ( thisBit ) new_angle [ servoNum ] = up_angle [
servoNum ]; // підняти
                    else new_angle [ servoNum ] = SERVOMIN; // опустити
                    servoNum ++; // Перейти до наступної серво

                }
            }
        }
    }

    // функція плавного повороту серво
void servoTick () {
    if ( turnFlag && IRFlag ) {
        if ( millis () - servoTimer > servoInterval ) {
            servoTimer = millis ();
            boolean waitFlag = false ;

            for ( byte i = 0; i < 28; i++) {
                if ( new_angle[i] > angle[i]) { // якщо кут більше
потрібного
                    waitFlag = true ;
                    angle[i] += servoStep ;
                    angle[i] = constrain ( angle[i], SERVOMIN, up_angle
[i]);

```

```

        if (i < 14) {
            hrs_drv.setPWM (i, 0, angle [i]);
        } else {
            mins_drv.setPWM (i - 14, 0, angle [i]);
        }
        // Serial.print (" up servo #"); Serial.print (i); Serial.print
(" "); Serial.println ( angle [i]);
    }
    else if ( new_angle [i] < angle [i]) { // якщо кут менше
потрібного
        waitFlag = true ;
        angle [i] -= servoStep ;
        angle [i] = constrain ( angle [i], SERVOMIN, up_angle
[i]);
        if (i < 14) {
            hrs_drv.setPWM (i, 0, angle [i]);
        } else {
            mins_drv.setPWM (i - 14, 0, angle [i]);
        }
        // Serial.print (" down servo #"); Serial.print (i);
Serial.print (" "); Serial.println ( angle [i]);
    } else if ( new_angle [i] == angle [i]) { // якщо кут дорівнює
потрібному
        // відключити серво
        if (i < 14) {
            hrs_drv.setPWM (i, 0, 4096);
        } else {
            mins_drv.setPWM (i - 14, 0, 4096);
        }
        if ( adj_flag && i == adj_servo ) {
// Тут корот "опустити і знову підняти індикатор під час команди
adj "
            adj_flag = false ;
            adj_flag2 = true ;
            new_angle [ adj_servo ] = up_angle [ adj_servo ];
        }
        // Serial.print (" servo #"); Serial.print (i); Serial.println
(" disabled ");
    }
    }
    turnFlag = waitFlag ;
}
}
// Тут корот "опустити і знову підняти індикатор під час команди
adj "- продовження
    if (adj_flag2) {
adj_flag2 = false ;
        turnFlag = true ;
    }
}
// опитувальник та парсер серіал
void serialTick () {
    if ( Serial.available () > 0) {
        String buf = Serial.readString ();

```



```

    if ( buf.startsWith ( " all-down ") ) {
        turnFlag = true ;
        for ( int i = 0; i < 28; i++) {
            new_angle [i] = SERVOMIN;
        }
        Serial.println (F(" Servos down "));
    }
    else if ( buf.startsWith ( " all-up ") ) {
        turnFlag = true ;
        for ( int i = 0; i < 28; i++) {
            new_angle [i] = up_angle [i];
        }
        Serial.println (F(" Servos up "));
    }
    else if ( buf.startsWith ( " set-time ") ) { // set-time 01:20
        int newH = buf.substring (9, 11). toInt ();
        int newM = buf.substring (12, 14). toInt ();
        if ( newH >= 0 && newH <= 23 && newM >= 0 && newM <= 59)
        {
            rtc.adjust ( DateTime (2012, 21, 12, newH , newM , 0));
            timeTimer = 60000 + millis ();
            Serial.print (F("RTC time set to "));
            Serial.print ( newH );    Serial.print (":");
Serial.println ( newM );
        } else {
            Serial.println (F(" Wrong) value !"));
        }
    }
    else if ( buf.startsWith ( " get-time ") ) {
        Serial.print (F("RTC time is "));
        DateTime now = rtc.now ();
        Serial.print ( now.hour ());    Serial.print (":");
Serial.println ( now.minute ());
    }
    else if ( buf.startsWith ( " down ") ) {
        int value = buf.substring (5). toInt ();
        if ( value >= 0 && value <= 28) {
            Serial.print (F(" Turning down servo #"));
            Serial.println ( value );
            new_angle [ value ] = SERVOMIN;
            turnFlag = true ;
        } else {
            Serial.println (F(" Wrong) value !"));
        }
    }
    else if ( buf.startsWith ( " up ") ) {
        int value = buf.substring (3). toInt ();
        if ( value >= 0 && value <= 28) {
            Serial.print (F(" Turning up servo #"));
            Serial.println ( value );
            new_angle [ value ] = up_angle [ value ];
            turnFlag = true ;
        } else {
            Serial.println (F(" Wrong) value !"));
        }
    }

```

```

}
}
else if ( buf.startsWith ( " speed ") ) {
    int value = buf.substring (6). toInt ();
    if ( value >= 0 && value <= 255) {
        servoStep = value ;
        EEPROM.writeByte (90, value );
        Serial.print (F(" Speed set to "));
        Serial.println ( servoStep );
    } else {
        Serial.println (F(" Wrong) value !"));
    }
}

else if ( buf.startsWith ( " adj ") ) { // adj 12 600
    byte num = buf.substring (4, 6). toInt ();
    int val = buf.substring (7). toInt ();
    up_angle [ num ] = val ; // встановити нове значення
    new_angle [ num ] = SERVOMIN;
    EEPROM.writeByte ( num * 2, val ); // записати на згадку
    Serial.print (F(" Adjust servo #")); Serial.print ( num
);

    Serial.print (F(" value ")); Serial.println ( val );
    turnFlag = true ;
    adj_flag = true ;
    adj_servo = num ;
}

if ( buf.startsWith ( " start-clock ") ) {
    Serial.println (F(" Clock start "));
    clock_state = true ;
}

if ( buf.startsWith ( " stop-clock ") ) {
    Serial.println (F(" Clock stop "));
    clock_state = false ;
}

if ( buf.startsWith ( " start-string ") ) {
    Serial.println (F(" Start string "));
    clock_state = false ;
    stringFlag = true ;
}

if ( buf.startsWith ( " stop-string ") ) {
    Serial.println (F(" Stop string "));
    clock_state = true ;
    stringFlag = false ;
}

if ( buf.startsWith ( " help ") ) {
    help ();
}
}
}
}

```