

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

(повне найменування інституту, назва факультету)

Кафедра програмованої електроніки, електротехніки і телекомунікацій

(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри  В. М. Рябенський

д.т.н., професор

« 16 » червня 2025 р.

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему: Розробка мікроконтролерної системи наливання нафтопродуктів у складі паливороздавального комплексу

Виконав: студент

4321 групи



А.Р. Піряник

(підпис)

Керівник роботи:

професор каф. ПЕЕТ, к.т.н., доцент,

професор НУК

(посада, науковий ступінь, вчене звання)



О. К. Жук

(підпис)

м. Миколаїв – 2025 року

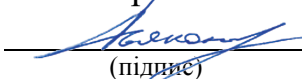
# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова  
Навчально-науковий інститут автоматики та електротехніки

|                  |                                                                    |
|------------------|--------------------------------------------------------------------|
| Кафедра          | <u>програмованої електроніки, електротехніки і телекомунікацій</u> |
| Спеціальність    | <u>171 «Електроніка»</u>                                           |
| Освітня програма | <u>Електронні системи</u>                                          |

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

 О. С. Дьяконов  
(підпис)

« 14 » квітня 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «бакалавр» (освітньо-кваліфікаційний рівень)

Студенту Пірянику Артуру Руслановичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка мікроконтролерної системи наливання нафтопродуктів у складі паливороздавального комплексу

керівник роботи Жук Олександр Кирилович, професор каф. ПЕЕТ, к.т.н., доцент, професор НУК  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ректора № УЧ-343 від «23» квітня 2025 року.

2. Термін подання роботи: 10 червня 2025 р.

3. Вихідні дані по роботі: розроблена програмна реалізація для управління відомчою паливороздавальною колонкою на базі мікроконтролера ATmega64.

Програмне забезпечення призначене для роботи з електронними модулями ПРК виробництва компанії «Шельф».

4. Перелік питань, що належать до розробки (найменування розділів): огляд існуючих систем управління ПРК; вибір платформи та визначення вимог до МКС; розробка програмного забезпечення для контролера ПРК; аналіз схемотехнічних рішень; створення модульної архітектури та реалізація алгоритмів керування; тестування і виправлення помилок; перевірка відповідності технічним вимогам; набуття практичних навичок програмування й діагностики ПРК; оцінка можливостей модернізації та самостійної розробки; аналіз ризиків і вимог охорони праці для безпечної роботи з електронікою.

5. Перелік презентаційних матеріалів: Стенд, використання логічного аналізатору, код програми реалізація керування, отримані навички, перспективи розвитку.

## 6. Консультанти розділів роботи

| Розділ   | Прізвище, ініціали та посада<br>Консультанта | Підпис, дата                                                                        |                                                                                     |
|----------|----------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|          |                                              | завдання<br>видав                                                                   | завдання<br>прийняв                                                                 |
| <b>5</b> | <b>Тубальцев А. М.</b>                       |  |  |
|          |                                              |                                                                                     |                                                                                     |
|          |                                              |                                                                                     |                                                                                     |
|          |                                              |                                                                                     |                                                                                     |

7. Дата видачі завдання «24» квітня 2025 р.

## КАЛЕНДАРНИЙ ПЛАН

| №<br>з/п | Назва етапів роботи                                       | Термін виконання<br>етапів роботи | Примітка |
|----------|-----------------------------------------------------------|-----------------------------------|----------|
| 1        | Ознайомлення з технічною літературою та нормативною базою | 02.02 - 18.02                     | виконано |
| 2        | Теоретичні основи та аналіз предметної області            | 19.02 - 03.03                     | виконано |
| 3        | Розробка ПЗ для контролера центральної плати ПРК          | 04.03 - 05.05                     | виконано |
| 4        | Тестування та перевірка працездатності                    | 06.05 - 12.05                     | виконано |
| 5        | Практичне значення роботи та шляхи розвитку               | 13.05 - 14.05                     | виконано |
| 6        | Охорона праці та техніка безпеки                          | 15.05 - 16.05                     | виконано |
| 7        | Написання вступу, висновків, оформлення списку літератури | 17.05 - 01.06                     | виконано |
| 8        | Коригування тексту за зауваженнями керівника              | 01.06 - 05.06                     | виконано |
| 9        | Підготовка доповіді, слайдів, презентації                 | 05.06 - 10.06                     | виконано |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |
|          |                                                           |                                   |          |

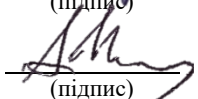
Студент

  
 (підпис)

А.Р. Піряник

(прізвище та ініціали)

Керівник роботи

  
 (підпис)

О. К. Жук

(прізвище та ініціали)

## АНОТАЦІЯ

Дипломна робота присвячена розробці програмного забезпечення для мікроконтролерної системи (МКС), що забезпечує автоматизоване керування процесом наливання нафтопродуктів у складі паливороздавального комплексу (ПРК).

У межах роботи використано апаратну частину, розроблену та виготовлену компанією «Шельф», яка є власником авторських прав на відповідні електронні модулі (центральна, силова та допоміжні плати). Програмну частину прошивки розроблено автором самостійно з урахуванням наявної схемотехніки та специфіки функціонування обладнання.

Окрему увагу в роботі приділено створенню тестового стенда, що дозволяє ефективно здійснювати налагодження, перевірку працездатності та демонстрацію роботи всієї мікроконтролерної системи. Стенд відтворює основні вузли ПРК і забезпечує умови, наближені до реальних, що дозволяє імітувати роботу обладнання без підключення до гідравлічної системи.

У першому розділі подано загальний огляд архітектури МКС ПРК, розглянуто вимоги до її функціонування, проведено аналіз існуючих систем автоматизації процесів дозування нафтопродуктів, а також обґрунтовано вибір апаратної платформи на базі мікроконтролера ATmega64.

Другий розділ присвячено розробці програмного забезпечення. В ньому детально описано логіку роботи системи, побудову програмних модулів, алгоритми керування процесом наливання палива та взаємодії між функціональними блоками. Розглянуто використання схемотехнічних рішень центральної, силової та допоміжних плат у процесі програмування.

У третьому розділі описано методику тестування прошивки, подано результати випробувань системи, проведено аналіз виявлених помилок та їх усунення, а також перевірено відповідність програмного забезпечення технічним вимогам.

Четвертий розділ акцентує увагу на практичному значенні виконаної роботи, а також на здобутих навичках у програмуванні мікроконтролерів, аналізі

схемотехніки та розробці ефективних керуючих алгоритмів. Наведено пропозиції щодо подальшої модернізації та створення власних апаратно-програмних рішень.

У п'ятому розділі розглянуто питання охорони праці, зокрема, електробезпеку, пожежну безпеку та вимоги до організації безпечного робочого місця розробника.

У результаті виконаної роботи створено повноцінну мікроконтролерну систему з індивідуально розробленим програмним забезпеченням, яка може бути використана у складі сучасних паливороздавальних комплексів для точного та надійного керування процесом наливання нафтопродуктів.

Ключові слова: мікроконтролерна система, контролер, АТтеда64, паливороздавальний комплекс, ПРК, нафтопродукти, автоматизація, прошивка, керування наливом, програмне забезпечення, схемотехніка, тестування, рідкокристалічний індикатор РКІ.

## ABSTRACT

This thesis is dedicated to the development of software for a microcontroller-based system (MCS) designed to automate the fuel dispensing process within a fuel dispensing complex (FDC).

The hardware components used in this work were developed and manufactured by the company "Shelf", which holds the intellectual property rights to the central, power, and auxiliary electronic modules. The firmware was developed entirely by the author, based on the existing circuit solutions and the functional specifics of the equipment. The system is implemented using the ATmega64 microcontroller.

Special attention in the work is paid to the creation of a test bench, which allows for effective debugging, performance testing and demonstration of the entire microcontroller system. The bench reproduces the main components of the control system and provides conditions close to real ones, which allows simulating the operation of the equipment without connecting to the hydraulic system.

The first chapter provides a general overview of the architecture of the MCS within the FDC, outlines the operational requirements, analyzes existing automation systems for petroleum product dispensing, and justifies the choice of the hardware platform.

The second chapter focuses on software development. It presents the system's operational logic, the structure of program modules, the control algorithms for fuel dispensing, and the interactions between functional blocks. The chapter also describes the integration of circuit-level solutions from the central, power, and auxiliary boards into the firmware.

The third chapter outlines the testing methodology, presents the test results, analyzes identified issues and solutions, and verifies compliance with technical specifications.

The fourth chapter highlights the practical significance of the developed system and the skills acquired in the process - such as microcontroller programming, schematic analysis, and the development of effective control algorithms. It also proposes potential

directions for future modernization and the creation of custom hardware-software solutions.

The fifth chapter addresses occupational safety, including electrical and fire safety considerations, and the requirements for organizing a safe and efficient developer workspace.

As a result of this work, a fully functional microcontroller system with custom-developed firmware has been created. It is capable of accurately and reliably controlling the fuel dispensing process within modern fuel dispensing complexes.

Keywords: microcontroller system, controller, ATmega64, fuel dispensing complex, FDC, petroleum products, automation, firmware, dispensing control, software, circuitry, testing, liquid crystal

## ЗМІСТ

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| Перелік умовних позначень, символів, одиниць, скорочень та термінів .....    | 9  |
| Вступ.....                                                                   | 11 |
| Розділ 1. Принципи побудови та удосконалення МКС у складі ПРК .....          | 16 |
| 1.1. Аналіз існуючих систем ПРК. Вибір апаратної платформи.....              | 16 |
| 1.2. Склад ПРК, структурна побудова та принцип дії.....                      | 24 |
| 1.3. Вимоги до МКС в ПРК .....                                               | 34 |
| 1.4. Постановка задачі та визначення вимог до програмного забезпечення ..... | 37 |
| Висновок до розділу 1 .....                                                  | 39 |
| Розділ 2. Розробка ПЗ контролера центральної плати ПРК.....                  | 42 |
| 2.1. Апаратна платформа мікроконтролера та програмні засоби розробки.....    | 42 |
| 2.1 Використання схемотехнічних рішень центральної плати .....               | 44 |
| 2.2 Програмна структура: модулі, функції та логіка роботи.....               | 50 |
| 2.3 Реалізація ключових алгоритмів керування паливною колонкою.....          | 55 |
| Висновки до розділу 2 .....                                                  | 62 |
| Розділ 3. Тестування та перевірка працездатності.....                        | 64 |
| 3.1. Методика тестування .....                                               | 64 |
| 3.2. Аналіз помилок та їх усунення .....                                     | 67 |
| 3.3. Результати випробувань .....                                            | 73 |
| 3.4. Перевірка відповідності технічним вимогам .....                         | 74 |
| Висновки до розділу 3 .....                                                  | 77 |
| Розділ 4. Практичне значення виконаної роботи та шляхи подальшого розвитку   | 79 |
| 4.1. Отримані навички в програмуванні мікроконтролерів.....                  | 79 |
| 4.2. Формування практичних навичок діагностики та ремонту ПРК.....           | 80 |
| 4.3. Можливості модернізації та створення власних електронних модулів .....  | 82 |
| 4.4. Перспективи самостійної розробки апаратно-програмних рішень .....       | 85 |
| Висновки до розділу 4 .....                                                  | 86 |
| Розділ 5. Охорона праці .....                                                | 89 |
| 5.1. Аналіз небезпечних факторів.....                                        | 89 |
| 5.2. Заходи щодо забезпечення електробезпеки .....                           | 92 |



|                                                            |     |
|------------------------------------------------------------|-----|
| 5.3. Пожежна безпека при експлуатації обладнання .....     | 95  |
| 5.4. Вимоги до організації робочого місця розробника ..... | 98  |
| Висновки до розділу 5 .....                                | 99  |
| Висновки .....                                             | 101 |
| Список використаних джерел .....                           | 103 |
| Додаток А .....                                            | 105 |
| Центральний код main.c.....                                | 105 |
| Додаток Б.....                                             | 132 |
| Асемблері функції для індикації .....                      | 132 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

|                    |                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| МКС                | – мікроконтролерна система                                                                                                                                                                                                            |
| ПРК                | – паливороздавальний комплекс                                                                                                                                                                                                         |
| ПЗ                 | – програмне забезпечення                                                                                                                                                                                                              |
| РКІ                | – рідкокристалічний індикатор                                                                                                                                                                                                         |
| ЕІО                | – Європейська центральна плата введення–виводу                                                                                                                                                                                        |
| FPGA               | – програмована користувачем матриця логічних елементів,<br>наприклад, інтегральна схема                                                                                                                                               |
| TCP/IP             | – керування передачею даних/протокол мережі інтернет                                                                                                                                                                                  |
| TQC                | – калькулятор Tokheim Quality Calculator                                                                                                                                                                                              |
| TQM                | – лічильник Tokheim quality meter                                                                                                                                                                                                     |
| SIO                | – керована плата вводу–виводу                                                                                                                                                                                                         |
| CAN                | – локальна мережа контролерів                                                                                                                                                                                                         |
| ECVR               | – система уловлювання пари з електронним управлінням                                                                                                                                                                                  |
| GPIO               | – вхід/вихід загального призначення                                                                                                                                                                                                   |
| LAN                | – локальна мережа                                                                                                                                                                                                                     |
| ATmega64           | – 8–бітний мікроконтролер архітектури AVR виробництва компанії Atmel, що має велику кількість портів введення/виведення, периферійні модулі, підтримку інтерфейсів UART, SPI, 12C та використовується для вбудованих систем керування |
| UART               | – Universal Asynchronous Receiver–Transmitter – універсальний асинхронний приймач–передавач                                                                                                                                           |
| RS–485             | – інтерфейс передачі даних на великі відстані, стійкий до перешкод                                                                                                                                                                    |
| Асемблер           | – мова програмування низького рівня для прямого управління апаратурою                                                                                                                                                                 |
| Асемблерна функція | – блок коду, написаний на асемблері, що може вбудовуватися в Сі–програму                                                                                                                                                              |

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Скорочення | Арк. |
| Змн. | Арк. | № документу | Підпис | Дата |                           | 9    |

Цикл – програмна конструкція для повторення дій (for, while)

Переривання – механізм реагування на зовнішні/внутрішні події без затримки

Таймер – модуль мікроконтролера для відліку часу

Компаратор – пристрій для порівняння двох аналогових сигналів

Симістор – електронний ключ, що використовується для керування змінним струмом (особливо в силовій частині)

Датчик пістолету (геркон) – датчик на основі геркона, фіксує зняття або вставлення пістолета в підпістолетник.

Датчик імпульсу – датчик, що зчитує кількість обертів дозатора для розрахунку об'єму пального

Гідравліка – сукупність вузлів і механізмів, що забезпечують рух палива в ПРК, включаючи клапани, дозатори, насоси

Порт – група виводів мікроконтролера для цифрового/аналогового вводу-виводу

Пін (вивід) – окрема фізична лінія (нога) мікроконтролера

Інтерфейс – засіб зв'язку між пристроями або програмними модулями

Стенд – відлагоджувальна макетна система, створена для тестування та демонстрації роботи МКС

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Скорочення | Арк. |
|      |      |             |        |      |                           | 10   |
| Змн. | Арк. | № документу | Підпис | Дата |                           |      |

## ВСТУП

У сучасних умовах стрімкого розвитку нафтогазової галузі зростають вимоги до точності, ефективності та безпеки всіх технологічних процесів. Автоматизація стає не просто бажаною, а необхідною складовою забезпечення стабільної роботи об'єктів паливно-енергетичного комплексу. Особливу роль у цій системі відіграють паливороздавальні комплекси, які відповідають за кінцеву ланку ланцюга постачання – безпосередній відпуск пального споживачу. Водночас, якщо гідравлічні вузли таких комплексів залишаються конструктивно незмінними впродовж десятиліть, то електронна складова активно розвивається, що відкриває широкі можливості для її оновлення та вдосконалення. Саме тому модернізація засобів керування ПРК шляхом впровадження сучасних мікроконтролерних систем є, на сьогодні, особливо актуальним завданням, яке дозволяє підвищити функціональність, надійність і технологічну гнучкість обладнання без необхідності повної заміни гідравлічної частини.

Значна частина паливороздавальних комплексів, які нині експлуатуються на території України, базується на електронних системах, розроблених ще у 1990–2000-х роках. Частина з них пройшла часткову модернізацію, але переважно це стосується лише заміни окремих блоків або адаптації під сучасні системи обліку. Водночас повноцінна заміна або оновлення програмно-апаратної частини для багатьох операторів є фінансово невигідною або технічно складною через відсутність документації, підтримки чи несумісність компонентів.

На українському ринку ПРК умовно поділяються на три категорії: високого, середнього та економ-класу. До сегменту високого рівня належать імпорتنі комплекси, зокрема Tokheim, Gilbarco Veeder-Root, Tatsuno Bench, які відзначаються широким функціоналом, високою надійністю та підтримкою сучасних протоколів зв'язку, проте є дорогими у встановленні та обслуговуванні. Середній сегмент представлений, зокрема, продукцією

|      |      |             |        |      |                      |      |
|------|------|-------------|--------|------|----------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Вступ | Арк. |
|      |      |             |        |      |                      | 11   |
| Змн. | Арк. | № документу | Підпис | Дата |                      |      |

вітчизняного виробника Шельф, а також системами з електронікою Хако (компанія "Геркон"), які поєднують задовільний рівень якості, сумісність із українською інфраструктурою та доступні витрати на підтримку. До бюджетного класу, що поступово втрачає актуальність, відносяться такі застарілі моделі, як Нара, Зельскотен, Глобалстар, Новопіньйон, які наразі практично не обслуговуються в Україні, а їх програмна та апаратна база не дозволяє інтегрувати сучасні методи керування або обліку без глибокої модернізації.

У таких умовах особливої актуальності набуває розробка мікроконтролерної системи керування, яка б могла інтегруватися в наявну апаратну базу та відповідала сучасним вимогам до масштабованості, гнучкості, надійності й точності. Основною метою цієї роботи є розробка програмного забезпечення для мікроконтролерної системи керування наливом палива, з використанням апаратної бази паливороздавального комплексу виробництва ТОВ «Шельф», а також реалізація отладного стенду для тестування й демонстрації роботи системи.

Об'єктом дослідження є процес керування відпуском нафтопродуктів у складі паливороздавального комплексу.

Предмет дослідження – структура, алгоритми та програмне забезпечення мікроконтролерної системи ПРК.

У ході дослідження були поставлені наступні завдання:

- провести аналіз існуючих систем керування ПРК;
- визначити вимоги до апаратної та програмної частини системи;
- розробити модульну архітектуру ПЗ для мікроконтролера ATmega64;
- реалізувати ключові алгоритми взаємодії з індикаторами, датчиками, виконавчими механізмами;
- створити тестовий стенд для перевірки функціонування системи;
- провести тестування та аналіз працездатності розробки.

|      |      |             |        |      |                      |      |
|------|------|-------------|--------|------|----------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Вступ | Арк. |
|      |      |             |        |      |                      | 12   |
| Змн. | Арк. | № документа | Підпис | Дата |                      |      |

Розробка виконувалась на основі апаратної бази, що застосовується у паливороздавальних комплексах виробництва компанії «Шельф», яка володіє авторськими правами на апаратну частину. У роботі використовувались плати, вилучені з обладнання після тривалої експлуатації, які були самостійно відремонтовані та приведені у повністю працездатний стан. У процесі проєктування програмного забезпечення мікроконтролера ATmega64 було враховано всі наявні схемотехнічні рішення центральної, силової та допоміжних плат.

При цьому жодних змін чи модифікацій до електричної схеми внесено не було — розробка виконувалась у повній відповідності до існуючої апаратної архітектури.

Також варто зазначити, що в більшості випадків, коли у центральній платі ПРК виходить з ладу мікроконтролер, виробник або сервісна організація не пропонує ремонт, а рекомендує заміну всієї плати на нову. Це суттєво підвищує вартість обслуговування та ускладнює експлуатацію обладнання в умовах обмежених бюджетів. Розробка власного програмного забезпечення для існуючої апаратної бази дозволяє продовжити строк служби таких плат та забезпечити гнучкі можливості їх модернізації.

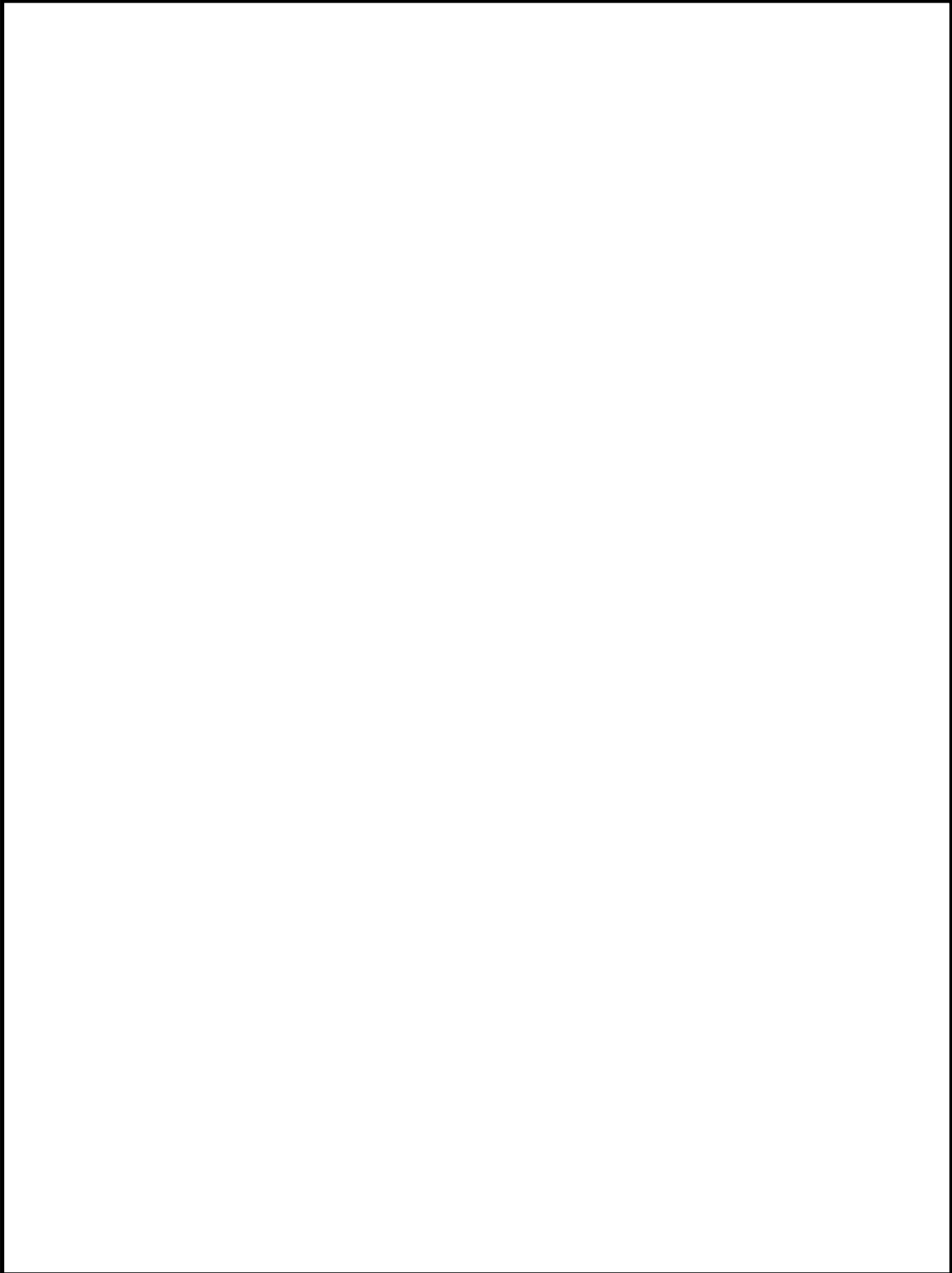
Розроблене програмне забезпечення дозволяє забезпечити базові функції керування та індикації, однак, у зв'язку з відсутністю доступу до захищених авторським правом протоколів передачі даних між ПРК та системами керування АЗС, реалізація повноцінної взаємодії з зовнішніми обліковими або платіжними системами не передбачена. Тому прошиті мікроконтролери можуть бути використані лише у некомерційних (відомчих) або експериментальних паливороздавальних комплексах.

Актуальність теми також обумовлюється необхідністю імпортозаміщення програмного забезпечення паливороздавальних комплексів, здешевленням технічного обслуговування автозаправних станцій та підвищенням технічної автономності вітчизняних розробок. З огляду на те,

|      |      |             |        |      |                      |      |
|------|------|-------------|--------|------|----------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Вступ | Арк. |
|      |      |             |        |      |                      | 13   |
| Змн. | Арк. | № документа | Підпис | Дата |                      |      |

що значна частина ПРК в Україні використовує електроніку, яка або частково застаріла, або має обмежену підтримку, виникає потреба у створенні надійних, адаптованих до існуючої апаратної бази рішень. Мікроконтролерні системи, зокрема на базі AVR-контролерів, як-от АТmega64, зарекомендували себе як перевірене, гнучке та економічно доцільне рішення для вбудованих застосувань у паливно-енергетичному секторі.

|      |      |             |        |      |                      |      |
|------|------|-------------|--------|------|----------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Вступ | Арк. |
|      |      |             |        |      |                      | 14   |
| Змн. | Арк. | № документу | Підпис | Дата |                      |      |



|             |      |                  |                                                                                     |      |                                                                               |                                |      |        |
|-------------|------|------------------|-------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------|--------------------------------|------|--------|
|             |      |                  |                                                                                     |      | 171.4321.КР.ПЗ.Р1                                                             |                                |      |        |
| Змн.        | Арк. | № документа      | Підпис                                                                              | Дата | Принципи побудови та<br>аналіз напрямків<br>удосконалення МКС у<br>складі ПРК | Літ.                           | Арк. | Аркуші |
| Розробник   |      | Піряник А.Р.     |  |      |                                                                               |                                | 15   |        |
| Консультант |      |                  |                                                                                     |      |                                                                               |                                |      |        |
| Н. контр.   |      | Добрінова Л. П.  |  |      |                                                                               |                                |      |        |
| Керівник    |      | Жук О. К.        |  |      |                                                                               |                                |      |        |
| Зав. каф.   |      | Рябенський В. М. |  |      |                                                                               | НУК<br>імені адмірала Макарова |      |        |



## РОЗДІЛ 1. ПРИНЦИПИ ПОБУДОВИ ТА АНАЛІЗ НАПРЯМКІВ УДОСКОНАЛЕННЯ МКС У СКЛАДІ ПРК

### 1.1. Аналіз існуючих систем ПРК. Вибір апаратної платформи

На сучасному українському ринку ПРК представлені широким спектром моделей, які умовно можна класифікувати за трьома категоріями: високого, середнього та бюджетного класу. Такий розподіл базується на сукупності факторів: функціональність, технічний рівень електроніки, підтримка сучасних протоколів зв'язку, рівень сервісної підтримки, а також вартість встановлення та обслуговування.

До преміального сегменту належать імпорتنі ПРК від таких провідних виробників, як Tokheim, Gilbarco Veeder-Root та Tatsuno Bench. Ці системи відзначаються високою точністю обліку, надійністю роботи в складних умовах, наявністю інтегрованих систем самообслуговування, зручними інтерфейсами користувача, а також повною підтримкою сучасних промислових протоколів, зокрема CAN-шини. Основним обмеженням їх широкого впровадження на ринку є висока вартість.

Серед ПРК високого класу одне з провідних місць займає продукція Tokheim[1] — всесвітньо відомого американського виробника паливороздавального обладнання. Обладнання цього бренду вирізняється інженерною досконалістю як апаратної, так і програмної частини, модульною архітектурою, високою надійністю, безперервністю роботи та широкими можливостями для масштабування.

Основним елементом керування у системі Tokheim TQC є плата ЕІО (рис.1.1), яка відповідає за обмін сигналами між усіма апаратними модулями: процесорною платою, гідравлічними модулями (до п'яти), комунікаційною платою, дисплейними модулями та допоміжною платою введення/виведення (SIO). Комунікація з дисплеями здійснюється через два CAN-інтерфейси. Логіка плати реалізована за допомогою програмованої логічної матриці

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 16   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

(FPGA), яка виконує широкий набір функцій: живлення дисплеїв, обробка цифрових входів/виходів (8 GPIO і 8 GPI), керування електродвигуном і клапанами в системі рекуперації парів (ECVR), обробка переривань з LAN та CAN-інтерфейсів, а також контроль джерел живлення й акумуляторної батареї.

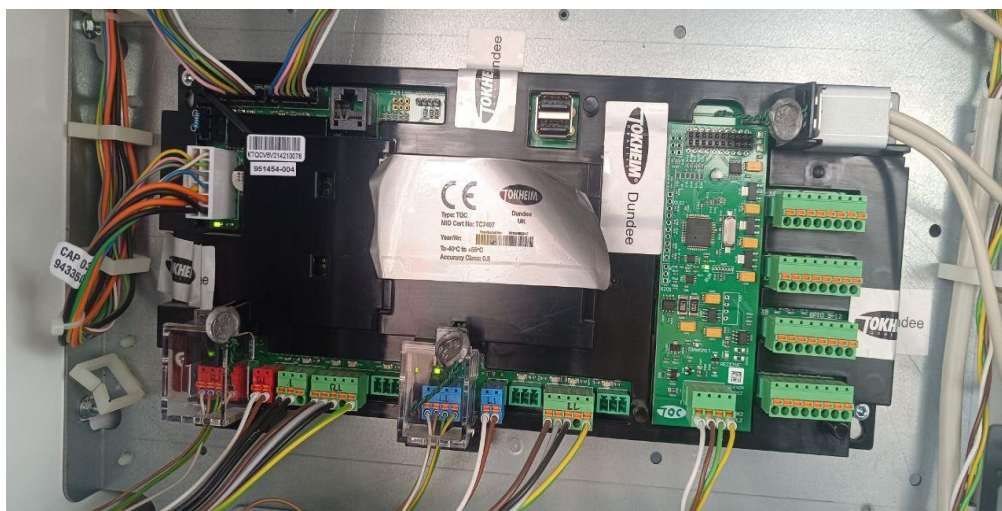


Рисунок 1.1 – Плата ЕІО

Центральною обчислювальною частиною комплексу є процесорна плата (рис.1.2), побудована на базі вбудованого ARM9-процесора AT91RM9200, яка працює під керуванням операційної системи Linux.

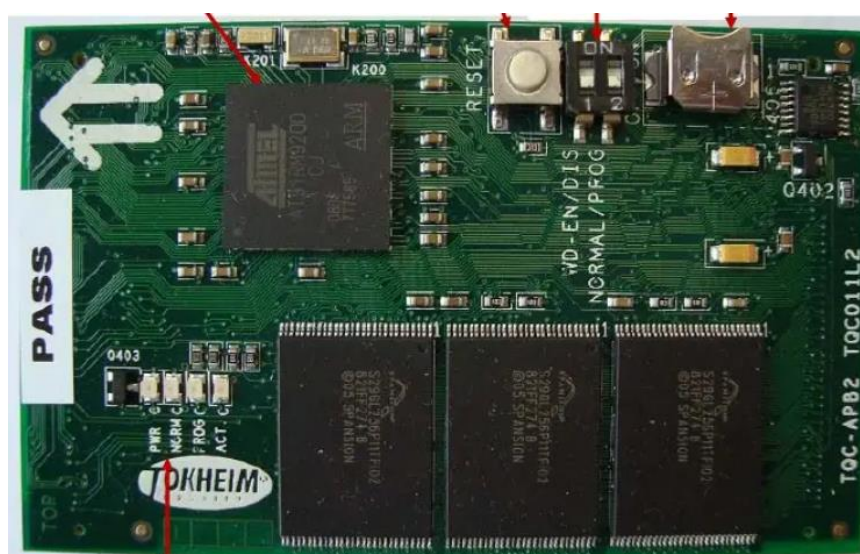


Рисунок 1.2 – Плата ARM

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.Р1 | Арк. |
| Змн. | Арк. | № документа | Підпис | Дата |                   | 17   |

Вона містить 64 МБ оперативної пам'яті SDRAM та 96 МБ флеш-пам'яті з можливістю розширення до 128 і 192 МБ відповідно. На платі також розміщено RTC (годинник реального часу) з резервним живленням, температурний датчик та мікросхему серійного номера. Живлення здійснюється через стабілізовані рівні 3,3 В і 1,8 В, що генеруються лінійними регуляторами під управлінням FPGA. Процесорна плата ініціалізує всю систему, запускає ОС, драйвери, прикладне ПЗ та протоколи обміну даними.

У порівнянні з ПРК середнього сегменту, де резервне живлення зазвичай реалізоване за допомогою іоністорів, у системах Tokheim використовуються акумуляторні батареї невеликої ємності. Це дає змогу забезпечити повноцінне завершення поточних операцій, збереження даних і контрольоване вимкнення у разі зникнення основного живлення, що значно підвищує надійність системи при роботі у нестабільних електромережах.

Система живлення плати ЕІО(рис.1.3) реалізована багаторівнево з розподілом по функціональних шинах: 8 В – для лінійних регуляторів та гідравлічних модулів; 5 В – для CAN-трансиверів і процесора; 3,3 В – для FPGA, препроцесора, інтерфейсів Ethernet та CAN; 1,2 В – для ядра FPGA; 24 В – для силових ліній GPIO.

Блок живлення Tokheim — це окремий спеціалізований пристрій, який виконує не лише функцію стабілізованого джерела живлення, а й працює як захисно-фільтруючий елемент. Він має багаторівневий захист від перевантаження по струму, напрузі, короткого замикання та перенапруги. У конструкцію пристрою входить потужний тороїдальний трансформатор, що згладжує імпульсні коливання вхідної напруги та компенсує її просідання. Крім того, передбачено окремі лінії живлення для цифрових і силових модулів, що мінімізує електромагнітні перешкоди та підвищує електромагнітну сумісність системи загалом.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 18   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

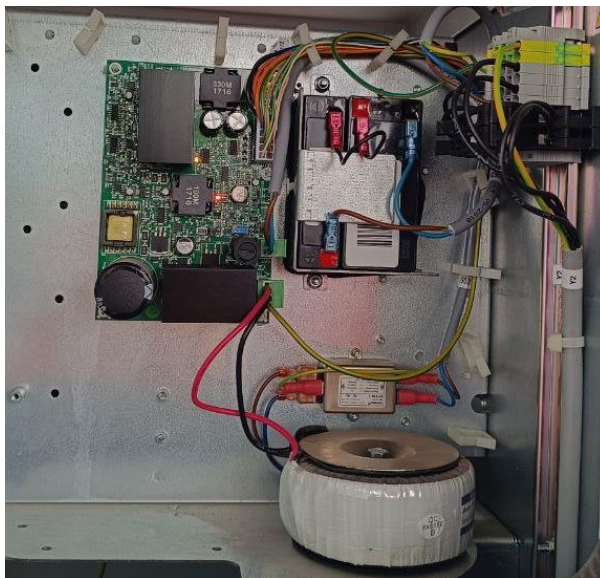


Рисунок 1.3 – Система живлення Tokheim

Гідравлічна частина Tokheim (рис.1.4) також має складну, але ефективну конструкцію. Вона включає високоточні дозатори з високою повторюваністю показників, електронні датчики тиску та витрати, цифрово-керовані зворотні клапани, а також систему температурної компенсації, яка забезпечує точний облік незалежно від температурних коливань середовища.



Рисунок 1.4 – Гідравлічний блок Tokheim

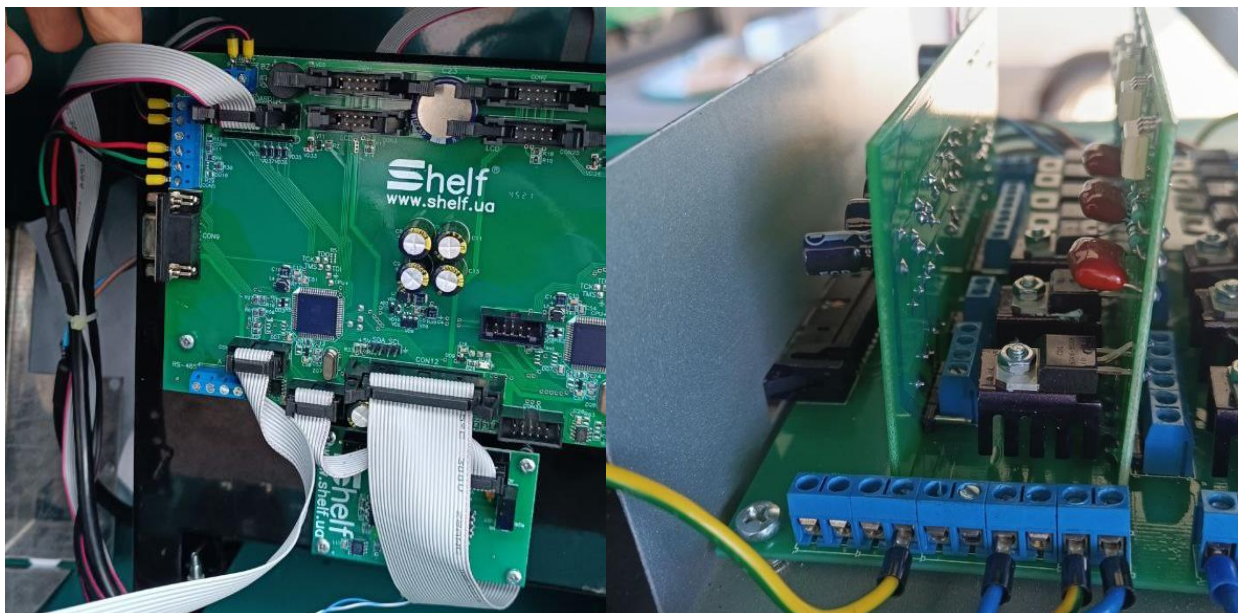
|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.Р1 | Арк. |
| Змн. | Арк. | № документу | Підпис | Дата |                   | 19   |



Загалом архітектура Tokheim відображає сучасний підхід до побудови інтелектуальних ПРК: усі компоненти мають чітку функціональну спеціалізацію, об'єднані через надійну CAN-шину, а кожен модуль є незалежною логічною одиницею, що спрощує обслуговування, модернізацію та локалізацію несправностей. Така багаторівнева система електроніки та гідравліки забезпечує безперервну, точну та безпечну роботу обладнання в умовах інтенсивної експлуатації.

До середнього класу ПРК належать вітчизняні системи, що поєднують задовільну точність, базову функціональність та доступну вартість. Такі рішення є оптимальними для невеликих АЗС, особливо у регіонах, де важливими є простота обслуговування, ремонтпридатність і доступність комплектуючих.

У цьому сегменті найчастіше зустрічаються розробки компанії «Шельф» (рис.1.5) [2], а також системи з електронікою ХАКО (компанія «Геркон») та «Slavutich». Вони мають просту мікроконтролерну архітектуру порівняно з системами високого класу, побудовану на основі AVR.



(а)

(б)

Рисунок 1.5 – Плати Шельф: (а)центральна плата ,(б)силовий блок

ПРК середнього рівня можуть бути як модульними, так і компактно інтегрованими. Наприклад, у системах «Шельф» використовується багатоплатна модульна архітектура з центральною логічною платою, окремим силовим блоком і платою індикації. Натомість системи ХАКО мають спрощену конструкцію[3], де центральна плата поєднує всю логіку, а силові елементи реалізовані на основі реле та магнітного пускателю (рис.1.6).

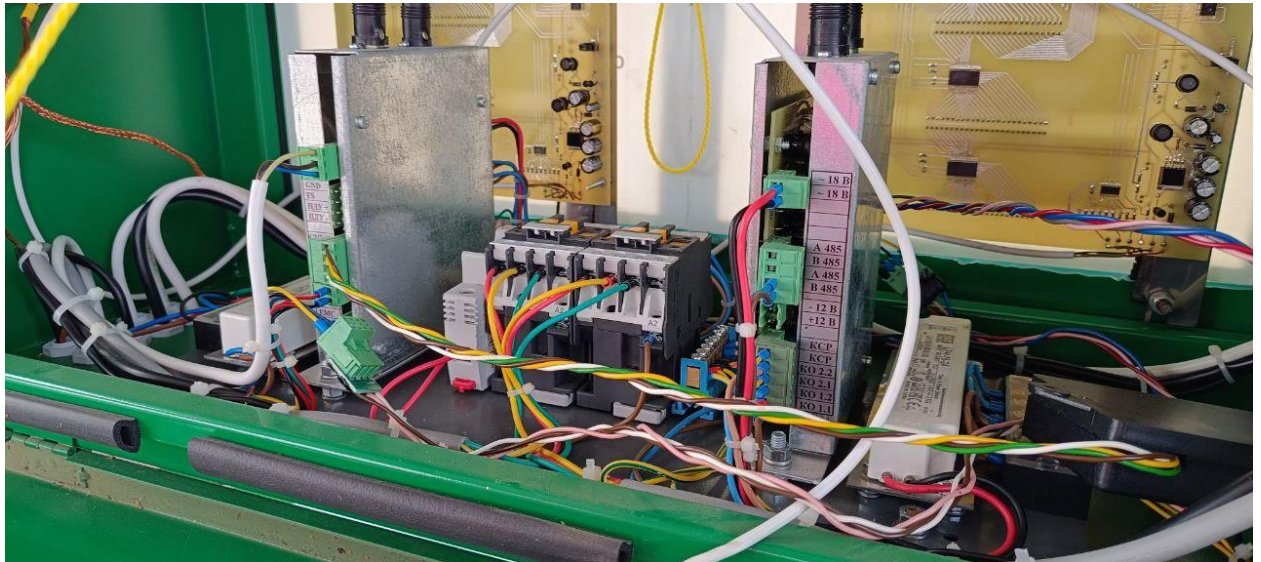


Рисунок 1.6 – Система ХАКО

Резервне живлення в таких системах зазвичай реалізується через іоністори, які дають змогу завершити налив та на деякий час залишити на РКІ останню інформацію . Це бюджетне, але ефективне рішення.

Гідравлічна система (рис.1.7) у паливороздавальних колонках середнього рівня зазвичай має стандартну та уніфіковану структуру, незалежно від виробника. Основні компоненти залишаються сталими як за конструкцією, так і за функціональним призначенням. До складу типового гідравлічного вузла входять: соленоїдний клапан, дозатор, насос, вибухозахищений двигун.



Рисунок 1.7 – Гідравлічна система Шельф

Паливороздавальні колонки середнього класу є оптимальним вибором для більшості невеликих або відомчих АЗС в Україні. Вони поєднують просту конструкцію, ремонтпридатність та низьку вартість обслуговування.

До бюджетної категорії належать переважно застарілі моделі паливороздавальних колонок, серед яких можна виділити такі як «Нара», «Зельскотен», «Глобалстар», «Новопіньйон» та інші. Ці системи були широко поширені у 1990–2000-х роках і свого часу забезпечували базові функції відпуску пального. Однак на сьогоднішній день їхнє використання обмежене через моральне та технічне старіння (рис. 1.8). Вони практично не обслуговуються в межах України, що ускладнює ремонт та підтримку працездатності. Апаратна частина таких колонок побудована на дискретних елементах або застарілих мікроконтролерах, а функціональні можливості значно поступаються вимогам сучасної інфраструктури АЗС. У більшості випадків спроба оновлення таких систем зводиться до повної заміни

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
| Змн. | Арк. | № документу | Підпис | Дата |                   | 22   |



керуючої електроніки. Таким чином, подальша експлуатація цих колонок можлива лише в умовах низького навантаження або на внутрішніх об'єктах без потреби інтеграції з АЗС-інфраструктурою.

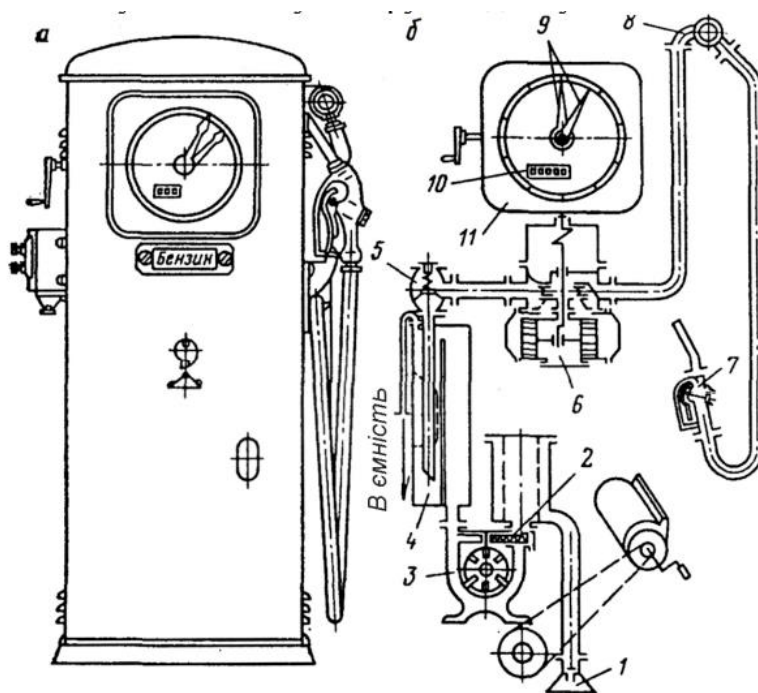


Рисунок 1.8 – Загальний вид (а) і технологічна схема (б) паливороздаточної колонки КЕР-40-0,5-1[4]

У рамках цієї роботи для дослідження та розробки програмного забезпечення було обрано паливороздавальну колонку середнього класу. Такий вибір є обґрунтованим з огляду на широку поширеність колонок цього сегменту на території України. Як сервісний інженер з обслуговування ПРК, я безпосередньо стикаюся з таким обладнанням у повсякденній практиці, тому маю реальний досвід діагностики, ремонту та модернізації систем саме цього класу.

Серед усіх рішень середнього класу для практичної реалізації було обрано саме систему «Шельф». Це зумовлено кількома ключовими факторами. По-перше, ПРК цього виробника є одними з найпоширеніших на ринку та мають модульну архітектуру, що спрощує їх діагностику,

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
| Змн. | Арк. | № документа | Підпис | Дата |                   | 23   |



обслуговування та модернізацію. По-друге, у межах моєї професійної діяльності в сервісній компанії маються неробочі модулі колонок «Шельф», які можна використати для створення навчально-тестового стенда. Це надало можливість не лише дослідити їхню будову, але й відновити працездатність шляхом власноручного ремонту, що є важливим практичним аспектом моєї роботи. Нарешті, вибір системи з зручною логікою побудови дозволив зосередитися саме на розробці програмного забезпечення, адаптованого до реальних умов експлуатації.

## 1.2. Склад ПРК, структурна побудова та принцип дії

Сучасні ПРК є складними електромеханічними системами, що поєднують гідравлічні, механічні та електронні компоненти (рис. 1.9). На далі для демонстрації будуть використані модулі «ШЕЛЬФ».

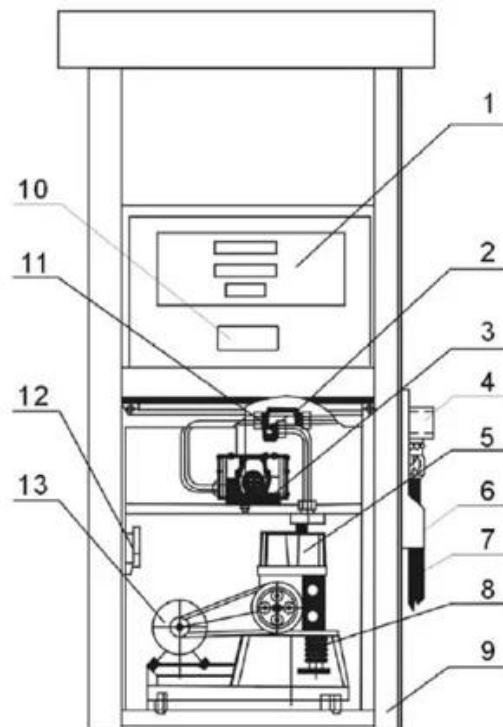


Рисунок 1.9– Структура ПРК:(1) Контролер,(2) Соленоїдний клапан,(3) Дозатор,(4) Індикатор нафтопродукту,(5) Насос,(6) Підпістолетник з датчиком,(7) Шланг нафтопродукту,(8) Гнучка труба (Сільфон),(9) Корпус із

покриттям,(10) Електромеханічний лічильник,(11) Імпульсний датчик,(12) Вибухозахищена розподільча коробка,(13) Вибухозахищений двигун [5]

Гідравлічний блок є ключовим компонентом ПРК (рис.1.10), що відповідає за механічну подачу, вимірювання й контроль за обсягом перекачаного пального. Його конструкція розрахована на тривалу експлуатацію у важких умовах, забезпечуючи надійність, точність та стійкість до впливу агресивного середовища.

У більшості ПРК використовується самовсмоктувальний лопатевий насос, який створює необхідний тиск для транспортування палива з резервуару до роздавального пістолета. Насос приводиться в дію електродвигуном, керування яким здійснюється через силову плату.

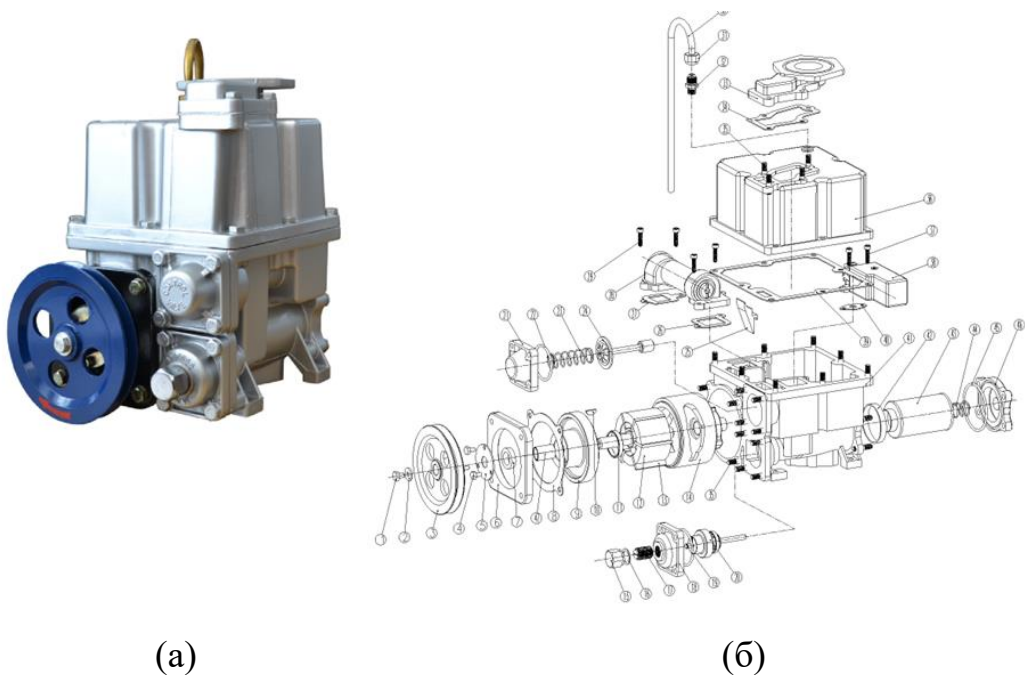


Рисунок 1.10 – Загальний вид насосу-моноблоку(а) та в розборі (б) [6]

Дозатор є найважливішим елементом гідравлічного блоку (рис. 1.11), що відповідає за точне вимірювання об'єму перекачаного пального. Як правило, використовується об'ємний дозатор з обертовим механізмом (наприклад, з овальними шестернями або роторним диском), який

приводиться в дію потоком пального. При кожному оберті дозатора видається фіксований об'єм рідини, що дозволяє дуже точно обчислювати кількість виданого пального.

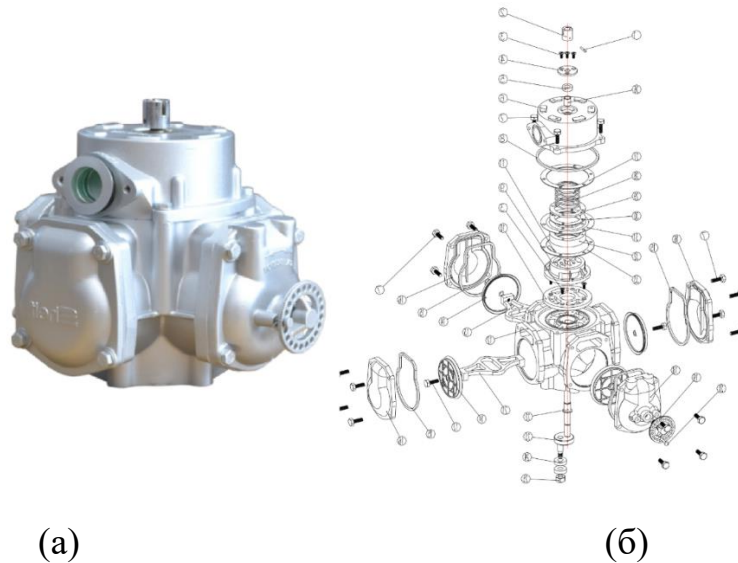


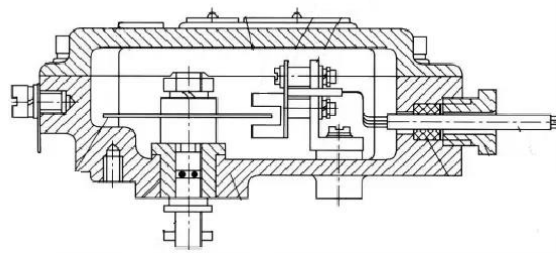
Рисунок 1.11 – Загальний вид дозатору(а) та в розборі (б)[6]

Імпульсний датчик є важливим електронним компонентом системи обліку об'єму пального(рис.1.12), який встановлюється безпосередньо на вихідному валу дозатора. Основним завданням датчика є перетворення механічного обертального руху дозатора на електричні імпульси, які надалі зчитуються мікроконтролерною системою для розрахунку кількості виданого пального.

Конструктивно датчик складається з перфорованого (дискретизованого) обертового колеса та щілинного оптичного сенсора. Під час обертання вала дозатора перфороване колесо обертається разом із ним, і коли проріз у колесі співпадає зі щілиною сенсора — формується електричний імпульс. Таким чином, за кожен оберт формується фіксована кількість імпульсів.



(a)



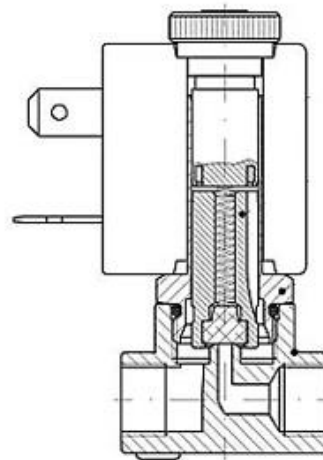
(б)

Рисунок 1.12 – Загальний вид імпульсного датчика(а) [7] та в розрізі (б) [8]

У розриві паливної магістралі встановлений електромагнітний (соленоїдний) клапан(рис.1.13), який відкривається для початку подачі пального та закривається для припинення. Клапан керується сигналами з мікроконтролера через силову плату. Подача живлення на котушку клапана відкриває прохід для пального. Надійність і швидкодія цього вузла важливі для точного завершення наливу.



(a)



(б)

Рисунок 1.13 – Загальний вид соленоїдного клапану(а) та в розрізі (б) [7]

Датчик пістолета (геркон) (рис. 1.14) – фіксує момент зняття/повернення роздавального пістолета.

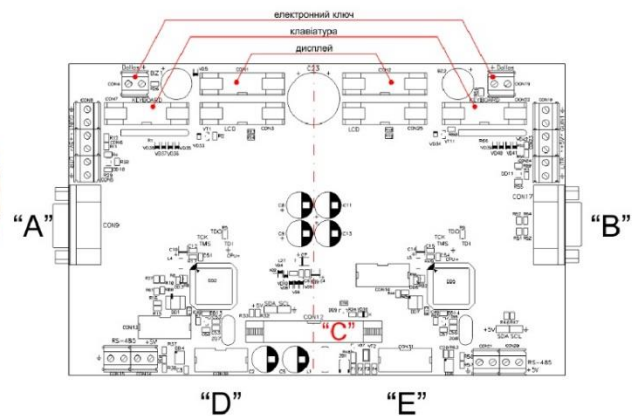


Рисунок 1.14 – Загальний вид підпістолетнику з датчиком пістолету[7]

Центральна плата керування є ядром мікроконтролерної системи паливороздавального комплексу (рис. 1.15), яка виконує основні логічні та комутаційні функції. Саме на цій платі розміщено мікроконтролер, що обробляє сигнали з датчиків, формує команди для виконавчих механізмів, керує індикацією та забезпечує обмін даними з периферійними модулями.



(а)

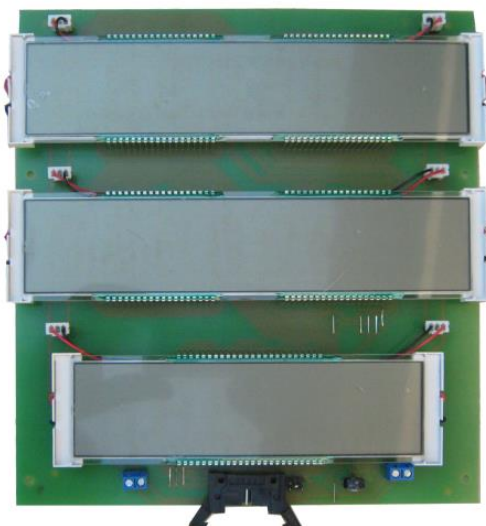


(б)

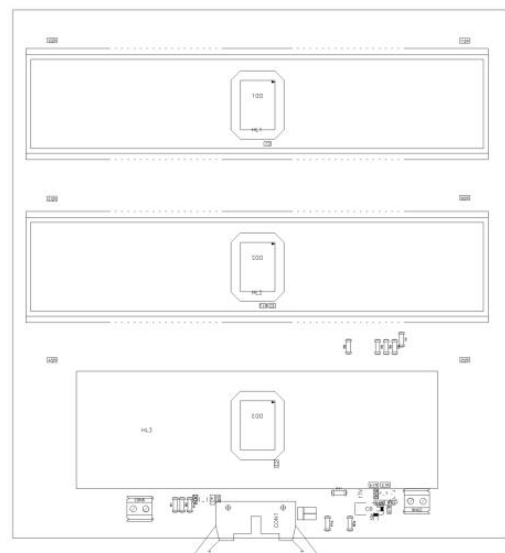
Рисунок 1.15 – Загальний вид центральної плати(а) та схема розташування елементів на платі (б) [7]

Плата індикації, або рідкокристалічний індикатор (РКІ) (рис. 1.16), є модулем для відображення основної інформації на паливороздавальному комплексі (ПРК). Вона виконує функцію зворотного зв'язку з оператором або клієнтом, наочно показуючи параметри процесу відпуску пального: кількість літрів, вартість, ціну за літр, та параметри налаштування .

У досліджуваній системі використовується сегментний РК-дисплей, що зібраний на основі драйверу NJU6445F. Ці контролери забезпечують мультиплексований вивід інформації на восьмисегментні цифри, що відображаються у трьох основних полях: сума до оплати ,об'єм відпущеного пального та ціна за літр.



(a)



(б)

Рисунок 1.16 – Загальний вид РК-дисплею (а) та схема розташування елементів на платі (б) [7]

Основними складовими силового блоку є плата контролю фаз, силова плата керування двигуном та клапанами, а також плата-підложка (монтажна база) (рис. 1.17). Плата контролю фаз призначена для перевірки наявності всіх трьох фаз електроживлення, що є критично важливим для стабільної роботи обладнання. Силова плата керування двигуном та клапанами є



основним елементом керування силовими ланцюгами; на ній реалізовано комутацію як електродвигуна паливного насоса, так і електромагнітних клапанів подачі палива. Ця плата містить силові ключі (симістори) та елементи гальванічної розв'язки, а керуючі сигнали надходять від центральної плати, яка визначає режим роботи системи на основі показань датчиків та заданих алгоритмів. За необхідності можливо підключити більше однієї плати на плату-підложку в залежності від задач. Плата-підложка виконує роль монтажною бази для розміщення силових плат та організації електричних з'єднань між модулями; на ній реалізовані доріжки для розподілу живлення, сигнальні лінії та роз'єми для підключення до інших частин системи.

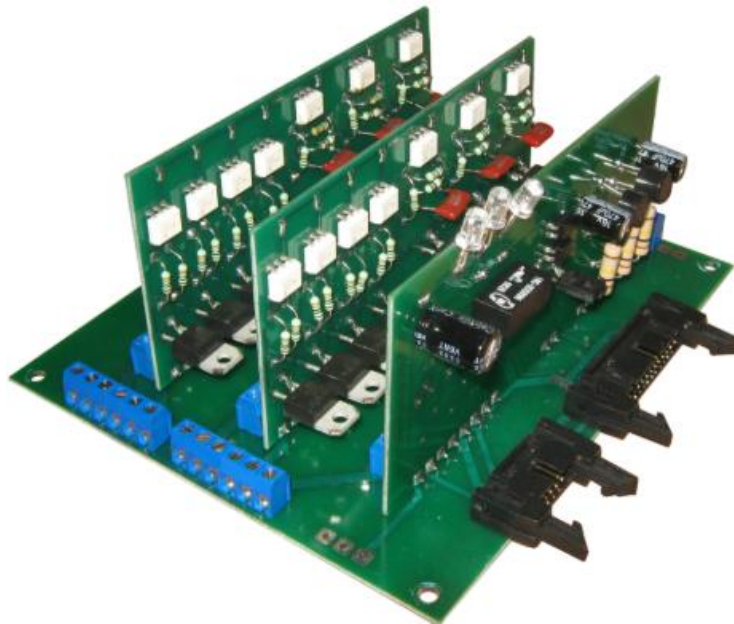


Рисунок 1.17 –Силовий блок і[7]

Інтерфейсна плата зв'язку виконує функцію передачі даних між центральною платою керування та зовнішніми модулями або системами керування (рис. 1.18). Вона є ключовою ланкою для організації надійного обміну інформацією в умовах промислових завод, що характерні для середовища роботи паливороздавального комплексу.

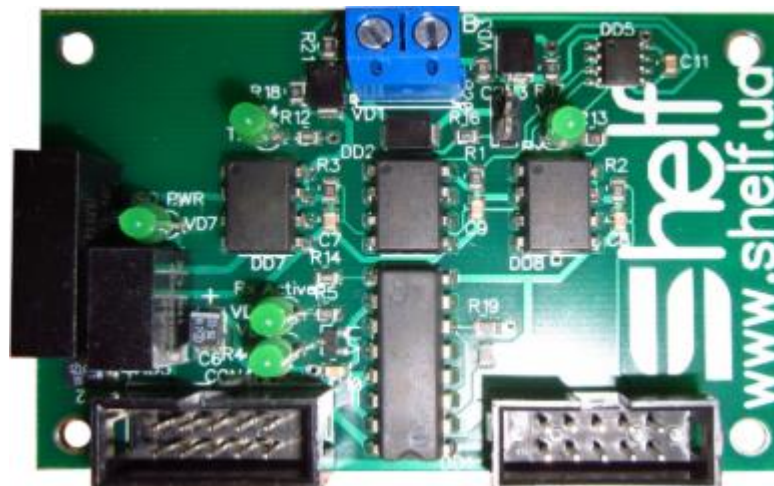


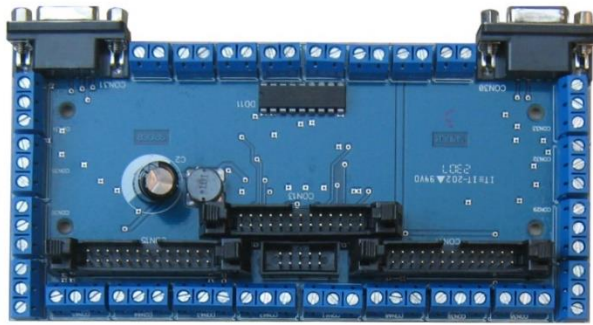
Рисунок 1.18 – Інтерфейсна плата RS485/UART[7]

У досліджуваному зразку використовується інтерфейс RS-485 — широко застосовуваний у промисловій автоматизації стандарт, що забезпечує диференційну передачу сигналів та дозволяє реалізувати стійкий до завад зв'язок на великі відстані (до 1200 м) із можливістю багатоточкової конфігурації (до 32 пристроїв на одній шині).

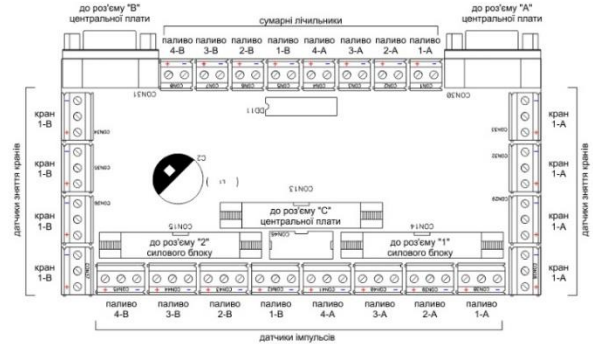
Інтерфейсна плата виконує низку важливих функцій: забезпечує гальванічне роз'єднання між логікою мікроконтролера центральної плати та зовнішньою шиною зв'язку, що підвищує електробезпеку та захищає систему від імпульсних перешкод або міжконтурних замикань; виконує буферизацію та підсилення сигналу за допомогою спеціалізованих драйверів, які дозволяють формувати стабільні диференційні сигнали навіть при значному навантаженні; здійснює передачу даних через апаратний UART мікроконтролера ATmega64, перетворюючи сигнали у формат RS-485 для надійної взаємодії з зовнішніми пристроями.

Плата комутації (рис. 1.19) – це опціональний модуль у складі ПРК, який застосовується у випадках, коли на одну сторону колонки підключено більше одного пістолета, або вся колонка обслуговує три й більше видів пального. Така ситуація потребує обробки сигналів від кількох незалежних каналів видачі палива.





(а)



(б)

Рисунок 1.19 – Загальний вид плата комутації (а) та схема розташування елементів на платі (б) [7]

Основне призначення цієї плати – збір, маршрутизація та уніфікація сигналів від герконів пістолетів та імпульсних датчиків, що контролюють оберти дозаторів, із подальшою передачею їх на центральну плату керування.

ПРК функціонує як автоматизована система керування подачею пального до транспортного засобу. Її робота базується на взаємодії електронного контролера з гідравлічними та електромеханічними компонентами (рис. 1.20).



Рисунок 1.20 – Структурна схема ПРК

Після ініціації процесу заправки – вручну через клавіатуру або через систему управління – контролер отримує сигнал на запуск. Контролер є головним обчислювальним блоком системи: він обробляє команди користувача, формує сигнали керування і здійснює контроль за станом усіх компонентів.

Контролер подає сигнал на силовий блок управління, який вмикає електродвигун. Обертальний момент з вала двигуна передається на вал насоса через ремінну передачу (рис. 1.21). Таким чином, насос починає створювати тиск у системі, закачуючи паливо з резервуара.

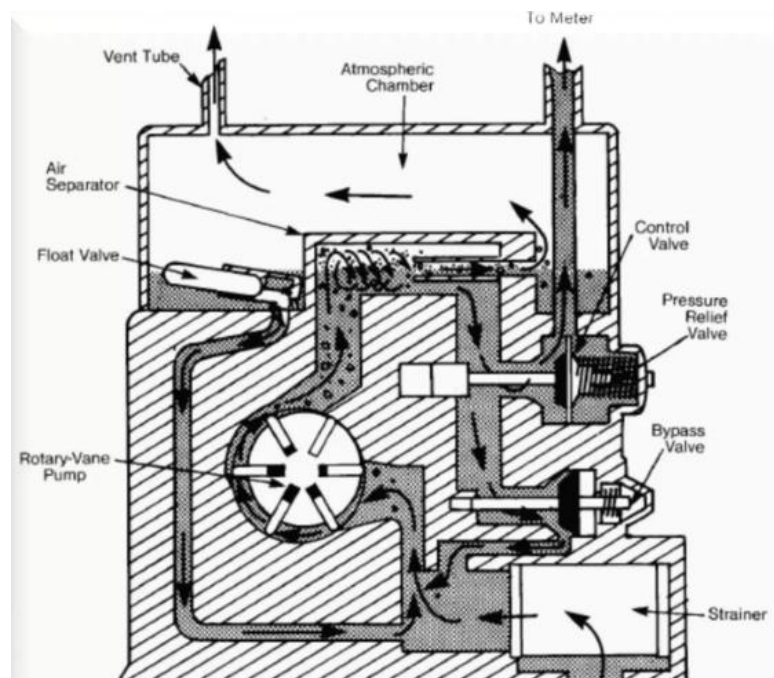


Рисунок 1.21 – Технологічна схема насосу-моноблоку[9]

Водночас з запуском мотору , контролер відкриває електромагнітний клапан, який у початковому стані закритий. Відкриття клапана дозволяє паливу проходити через дозатор, який забезпечує точне вимірювання об’єму пального, що подається. Дозатор виконує функцію витратоміра і пов’язаний із датчиком імпульсів — кожне обертання чи переміщення внутрішніх елементів дозатора генерує електричний імпульс, що надходить до контролера.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
| Змн. | Арк. | № документа | Підпис | Дата |                   | 33   |

На основі цих імпульсів контролер обчислює фактичний об'єм відпущеного палива та відповідну вартість. Результати виводяться на дисплеї.

У пістолет вбудований магніт, а в підпістолетник геркон, який при замиканні інформує контролер про початок і завершення видачі пального.

Після досягнення заданої кількості палива або при ручному завершенні заправки контролер надсилає команду на вимкнення електромагнітного клапана і двигуна, припиняючи подачу пального.

### 1.3. Вимоги до МКС в ПРК

Мікроконтролерна система виконує функції управління основними вузлами ПРК, забезпечуючи точне дозування пального, індикацію, керування виконавчими елементами та зв'язок із зовнішніми пристроями. Вимоги до МКС залежать від цільового призначення системи, і умовно їх можна поділити на два типи.

Для використання у ПРК комерційного призначення (рис. 1.22), повинна відповідати підвищеним вимогам до надійності, точності, енергонезалежності та сумісності з системами обліку. Оскільки ПРК використовуються для комерційного обліку відпуску пального, кожен етап дозування повинен суворо контролюватися та документуватися, а сама МКС має забезпечувати безперебійну роботу в режимі 24/7. Однією з ключових вимог є відповідність метрологічним стандартам — МКС повинна забезпечувати точну обробку імпульсів від датчиків витрати, гарантувати фіксацію об'єму з мінімальною похибкою та забезпечувати можливість калібрування. Для цього використовуються високоточні імпульсні датчики у поєднанні з програмними алгоритмами компенсації температури та тиску.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 34   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



Рисунок 1.22 – Комерційне АЗС «Кворум-нафта»

Також важливо, щоб система мала функцію фіксації останньої операції у разі втрати живлення. З цією метою МКС повинна бути обладнана резервним джерелом живлення (акумулятором або іоністорами), яке забезпечить завершення дозування. Ще одна важлива вимога — підтримка сучасних протоколів зв'язку, зокрема RS-485 або протоколів, сумісних з АЗС-обліком. МКС має мати можливість роботи у складі автоматизованих систем управління АЗС, забезпечуючи передачу даних у реальному часі про відпуск, стан вузлів, облікові параметри тощо (рис. 1.23).



Рисунок 1.23 – Програма управління ПРК на АЗС «АМІС»

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 35   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

У контексті програмного забезпечення МКС повинна мати реалізований захист від збоїв та адаптації під специфіку конкретної АЗС. Бажано, щоб вона підтримувала інтерфейс користувача для технічного налаштування та роботи з сервісними даними. Завдяки цим функціональним та конструктивним вимогам, МКС у ПРК комерційного призначення забезпечує стабільну та безпечну роботу у складних умовах експлуатації та відповідає нормативним вимогам до засобів комерційного обліку палива.

Для використовуються у відомчих або некомерційних ПРК(рис.1.24), мають менш жорсткі вимоги порівняно з комерційними аналогами, проте повинні залишатися надійними, простими в обслуговуванні та економічно доцільними. Основна функція таких систем — забезпечити контроль за відпуском палива для внутрішніх потреб підприємства, без обов’язкового проходження державної метрологічної перевірки. Це дозволяє спростити архітектуру, зменшити витрати на виробництво та обслуговування, і тим самим зробити систему доступнішою для підприємств із власним автопарком.

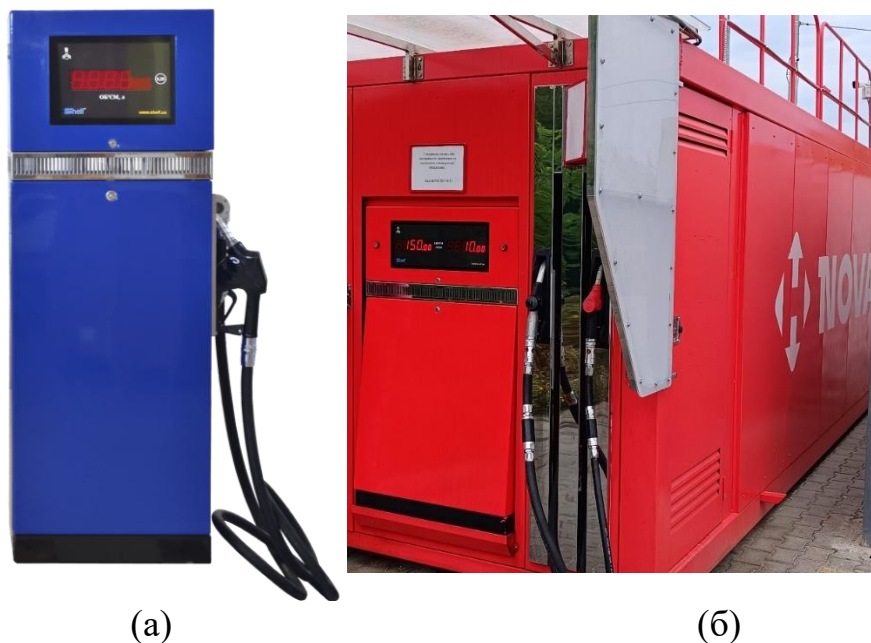


Рисунок 1.24 – (а)Відомча колонка Шельф[10](б)ПРК в депо Нова пошта

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 36   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

МКС відомих ПРК повинна підтримувати базові функції обліку: фіксацію обсягу пального, контроль прав доступу, обмеження за обсягом або часом дозування, а також збереження історії відпусків. Найчастіше такі системи працюють автономно, без підключення до зовнішніх облікових чи бухгалтерських систем, або ж підтримують спрощені протоколи обміну по інтерфейсу RS485 для передачі даних на локальний комп'ютер .

Таким чином, мікроконтролерні системи паливороздавальних колонок розробляються з урахуванням специфіки їхнього застосування — як у комерційній, так і у відомчій сферах. У комерційних ПРК МКС мають забезпечувати високу точність, підтримку метрологічних вимог, захищеність даних, сумісність із зовнішніми обліковими системами та стабільну роботу в умовах великого навантаження. Водночас для некомерційного (відомчого) використання ключовими є простота, ремонтпридатність, модульність та економічна доцільність. Обидва типи систем мають спільну мету — забезпечити контроль за відпуском палива, проте різняться ступенем складності, функціональністю та вартістю. Такий підхід дозволяє адаптувати ПРК під конкретні завдання, оптимізуючи витрати без шкоди для ефективності та надійності експлуатації.

#### **1.4. Постановка задачі та визначення вимог до програмного забезпечення**

У межах даної роботи основним завданням є розробка програмного забезпечення для мікроконтролерної системи паливороздавальної колонки середнього класу, зокрема на базі обладнання компанії «Шельф». З огляду на специфіку експлуатації таких колонок — переважно у складі регіональних автозаправних станцій або внутрішньовідомих об'єктів — програмне забезпечення має забезпечити стабільне, надійне та ефективне управління усіма етапами процесу дозування пального.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 37   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

Програмна частина має реалізовувати основні функціональні блоки ПЗ, зокрема: ініціалізацію периферійних пристроїв (дисплей, клавіатура, датчики), зчитування сигналів із датчика, керування електромагнітним клапаном та двигуном насоса, логіку розрахунку об'єму відпущеного пального, формування сигналів на індикацію, а також збереження критичних даних у разі аварійного вимкнення живлення.

У цій роботі програмне забезпечення розробляється виключно для відомчої паливороздавальної колонки. Це зумовлено закритістю та недоступністю протоколів передачі даних, що використовуються у комерційних системах керування та взаємодії з ПРК. Повноцінна реалізація програмної частини для таких колонок вимагала б доступу до закритої документації виробника, що неможливо в межах даної дипломної роботи. Відомчі ж системи не потребують сертифікованого комерційного обліку й допускають реалізацію відкритої або адаптованої логіки керування, що робить їх ідеальними для навчальних, експериментальних та сервісних завдань.

До програмного забезпечення висуваються конкретні функціональні та технічні вимоги, які визначають його надійність, точність і відповідність завданням відомчої паливороздавальної колонки.

Передусім необхідно забезпечити точне та своєчасне відображення інформації на рідкокристалічному індикаторі. Усі значення — обсяг виданого пального, стан системи, службові повідомлення — повинні миттєво оновлюватися та бути зручними для зчитування оператором.

Ключовим завданням є правильна обробка сигналів з імпульсного датчика. Програма повинна коректно рахувати імпульси, що відповідають кількості відпущеного пального, з урахуванням можливих перешкод, завад або шумів. Це необхідно для забезпечення високої точності обліку.

Також важливо забезпечити точну обробку сигналу з датчика пістолета. Програмне забезпечення має виявляти момент зняття або

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 38   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



повернення пістолета на місце, відповідно запускаючи або зупиняючи процес дозування пального, блокуючи подачу у разі нештатної ситуації.

Одним з основних завдань є правильне керування силовим блоком управління. Необхідно реалізувати логіку запуску та зупинки насоса, відкриття та закриття електромагнітного клапана, дотримуючись встановленої послідовності та безпечних затримок.

Програмне забезпечення повинне реалізовувати повний цикл видачі пального: зняття пістолета, запуск насоса, дозування пального з точним підрахунком, завершення сеансу та збереження результатів у пам'яті.

Усі ці процеси повинні відбуватись у режимі реального часу, без затримок та збоїв, що критично для забезпечення стабільної роботи системи.

Особливу увагу слід приділити надійності та відмовостійкості — у разі раптового вимкнення живлення програма повинна коректно завершити активний сеанс, зберігши основні дані до енергонезалежної пам'яті.

Також важливо забезпечити ефективне використання ресурсів мікроконтролера — оперативної пам'яті, таймерів, портів вводу/виводу. Код має бути оптимізованим для роботи на архітектурі ATmega64.

Крім цього, бажано дотримуватись модульного підходу у розробці, щоб у майбутньому можна було легко вносити зміни або розширювати функціональність системи без повної переробки програмного забезпечення.

## **Висновок до розділу 1**

У результаті аналізу існуючих систем паливороздавальних колонок (ПРК) було визначено, що для реалізації програмного забезпечення доцільно обрати систему середнього класу, яка поєднує достатню функціональність, поширеність в Україні та ремонтпридатність. Серед таких систем найбільш доцільним виявився вибір ПРК типу «ШЕЛЬФ» завдяки її модульній архітектурі, широкій доступності комплектуючих і можливості відновлення наявних вузлів.

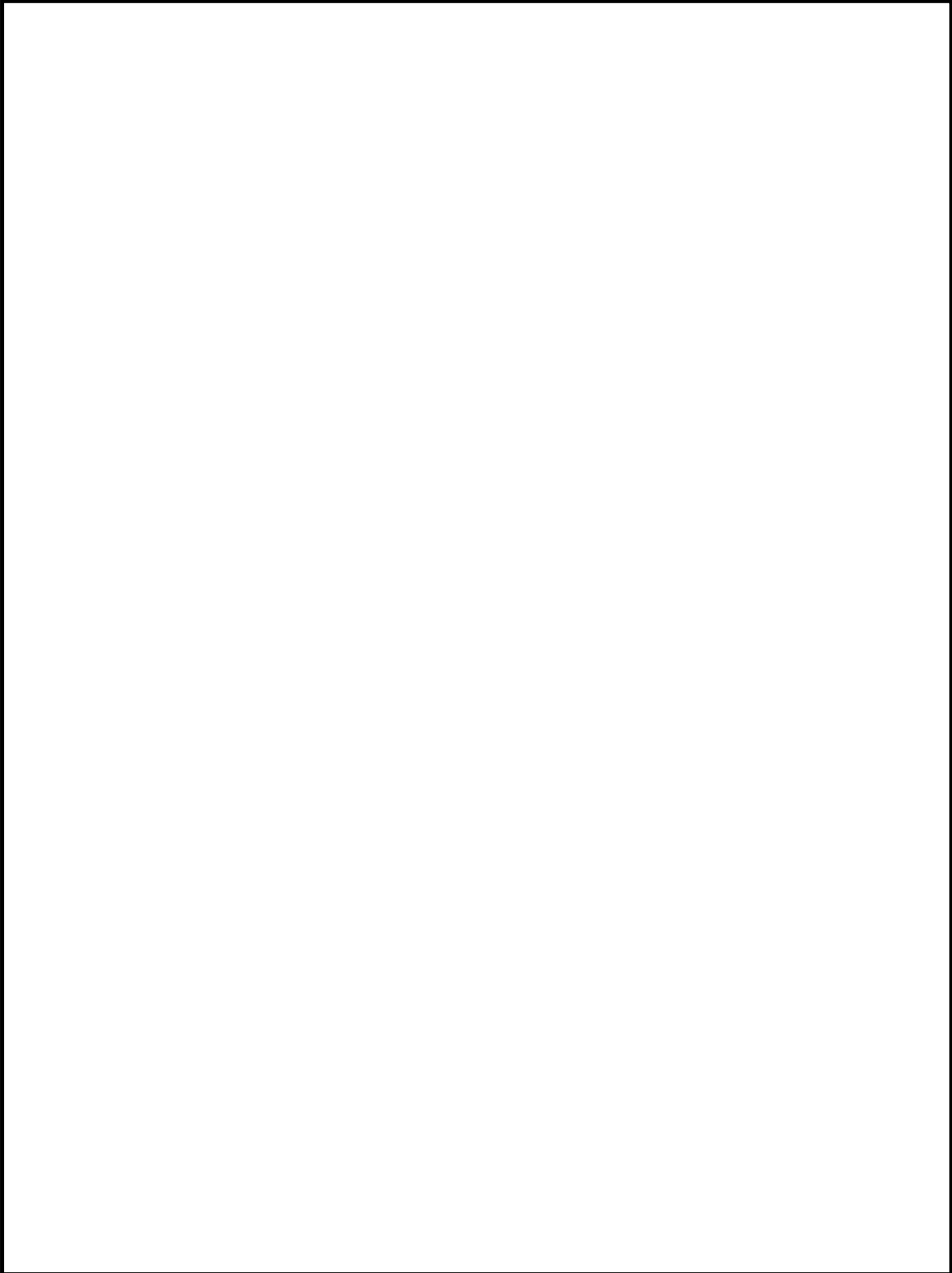
|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 39   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

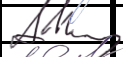
Огляд апаратної структури та принципу дії ПРК «ШЕЛЬФ» дозволив детально зрозуміти функціональну взаємодію між елементами системи — контролером, насосом, дозатором, індикатором та периферією, що стало основою для формування вимог до мікроконтролерної системи.

Під час аналізу вимог до МКС було встановлено, що система має забезпечувати точний облік пального, надійне управління силовими елементами та стабільну роботу з усіма датчиками. При цьому важливо враховувати різницю між комерційними та відомчими ПРК — у контексті цієї роботи реалізується саме відомчий варіант, орієнтований на внутрішній облік пального.

Постановка задачі для програмного забезпечення охоплює чіткі функціональні вимоги: обробку сигналів з імпульсного датчика та датчика пістолета, точне керування силовим блоком, оперативне відображення даних на індикаторі та забезпечення роботи в реальному часі. Сформульовані вимоги стануть основою для розробки ефективної, надійної та оптимізованої програми для керування ПРК «ШЕЛЬФ» на базі мікроконтролера ATmega64.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р1 | Арк. |
|      |      |             |        |      |                   | 40   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



|             |      |                  |                                                                                     |      | 171.4321.КР.ПЗ.Р2                                                      |                                |      |          |
|-------------|------|------------------|-------------------------------------------------------------------------------------|------|------------------------------------------------------------------------|--------------------------------|------|----------|
|             |      |                  |                                                                                     |      |                                                                        |                                |      |          |
| Змн.        | Арк. | № документа      | Підпис                                                                              | Дата | Розробка програмного забезпечення для контролера центральної плати ПРК | Літ.                           | Арк. | Аркушіів |
| Розробник   |      | Піряник А.Р.     |  |      |                                                                        |                                |      |          |
| Консультант |      |                  |                                                                                     |      |                                                                        |                                | 41   |          |
| Н. контр.   |      | Добрінова Л. П.  |  |      |                                                                        | НУК<br>імені адмірала Макарова |      |          |
| Керівник    |      | Жук О. К.        |  |      |                                                                        |                                |      |          |
| Зав. каф.   |      | Рябенський В. М. |  |      |                                                                        |                                |      |          |

## РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОНТРОЛЕРА ЦЕНТРАЛЬНОЇ ПЛАТИ ПРК

### 2.1. Апаратна платформа мікроконтролера та програмні засоби розробки

У даній роботі як апаратну платформу обрано мікроконтролер ATmega64(рис.2.1), який належить до сімейства AVR виробництва компанії Microchip.

Цей мікроконтролер має 64 КБ флеш-пам'яті програм, 4 КБ оперативної пам'яті SRAM, 2 КБ енергонезалежної пам'яті EEPROM, а також широкий набір периферійних модулів. До його складу входять чотири таймери (два 8-розрядні та два 16-розрядні), апаратні інтерфейси UART, SPI, I<sup>2</sup>C (TWI), аналоговий компаратор, 10-розрядний АЦП з кількома каналами, що дозволяє здійснювати точний контроль аналогових сигналів. Робоча напруга живлення становить від 4.5 до 5.5 В. Завдяки своїм характеристикам, ATmega64 повністю задовольняє базові потреби системи управління паливороздавальною колонкою.

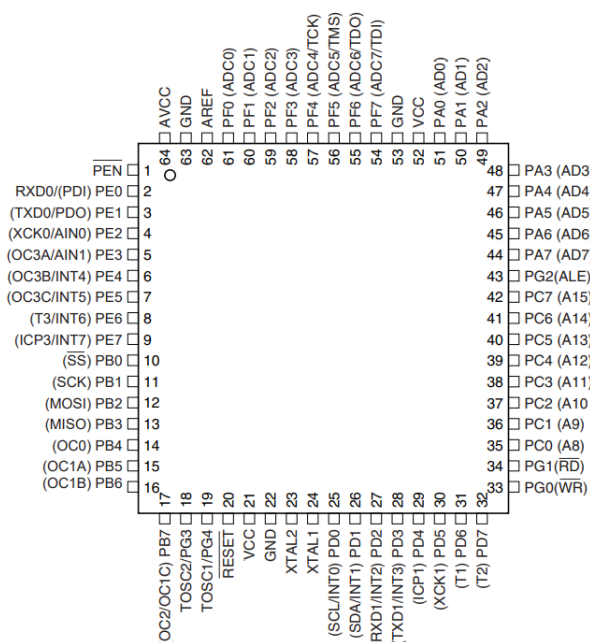


Рисунок 2.1 – Розпіновка Atmega64[11]

The screenshot shows the Qt Creator IDE interface. The main editor displays a C source file named `main.c`. The code defines several macros for bit manipulation and implements a `main()` function that configures hardware registers.

```

1776 int main() {
1777 |
1778 |   DORDI = (1 << HOT) ;
1779 |   DDRC = (1 << VCUT) ;
1780 |   DORG = (1 << VLOW) ;
1781 |
1782 |   PORTO =(1 << HOT) ;
1783 |   PORTC=(1 << VCUT) ;
1784 |   PORTO =(1 << VLOW) ;
1785 |
1786 |
1787 |   DDRA |= (1 << DATA_PIN) | (1 << SCK_PIN) | (1 << BUZ2) ;
1788 |   PORTA |= ~(1 << DATA_PIN) | (1 << SCK_PIN) | (1 << BUZ2)) ;
1789 |
1790 |   DDRA |= ~(1 << STR1 ) | (1 << STR2) | (1 <<STR3)|(1 <<STR4 )) ;
1791 |   PORTA |= ~(1 <<STR1 ) | (1 << STR2) | (1 <<STR3)|(1 <<STR4 )) ;
1792 |
1793 |   DDRF |= (1 << KOL1) | (1 <<KOL2 ) | (1 << KOL3) | (1 <<KOL4 ) ;
1794 |   PORTF |= ~(1 << KOL1) | (1 << KOL2) | (1 << KOL3) | (1 << KOL4 )) ;
1795 |
1796 |   DDRE |= (1 << PSQ2) ;
1797 }
  
```

On the right side, the "Solution Explorer" pane lists the project files:

- `call_nums.c`
- `call_nums.h`
- `call_x05.c`
- `call_x05.h`
- `call_xm5.c`
- `call_xm5.h`
- `call_hand5.c`
- `call_hand5.h`
- `call_memu5.c`
- `call_numbers.c`
- `call_nums.c`
- `call_nums.h`
- `call_ports.c`
- `call_ports.h`
- `call_ready5.c`
- `call_ready5.h`
- `call_xandmemu5.c`
- `call_xandmemu5.h`
- `call_xm5.c`
- `call_xm5.h`
- `main.c`

The bottom status bar indicates the current build configuration: "Build + IntelliSense".

Прошивання мікроконтролера виконувалося за допомогою популярного програматора USBasp (рис. 2.3), який дозволяє проводити запис через стандартний інтерфейс ISP. Для безпосереднього завантаження прошивки використовувалась окрема утиліта AVRDUDE\_PROG, яка забезпечує гнучке налаштування параметрів програмування.



|             |             |                    |               |             |                   |      |
|-------------|-------------|--------------------|---------------|-------------|-------------------|------|
|             |             |                    |               |             | 171.4321.КР.ПЗ.Р2 | Арк. |
|             |             |                    |               |             |                   | 43   |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ документа</i> | <i>Підпис</i> | <i>Дата</i> |                   |      |

## 2.1 Використання схемотехнічних рішень центральної плати

Табло для відображення інформації складається з трьох рідкокристалічних дисплеїв: двох вісьмисимвольних та одного шестисимвольного, які використовуються для індикації кількості виданого пального, суми та ціни за літр. Кожен з дисплеїв керується окремим драйвером NJU6445F. Ці драйвери з'єднані послідовно між собою, тому передача даних реалізована за принципом зсувного регістра, де інформація спочатку надходить до останнього драйвера, а потім проходить до попередніх. Передача даних до драйверів здійснюється з мікроконтролера ATmega64 через три сигнальні лінії: PA6 використовується для передавання даних, PA7 – для тактового сигналу, а PG2 – як сигнал завершення передачі. Після надсилання необхідної кількості бітів PG2 активується, що сигналізує драйверам зафіксувати отриману інформацію. Такий підхід дозволяє ефективно керувати кількома дисплеями з мінімальною кількістю задіяних виводів мікроконтролера. Індикація з'являється в порядку зворотному до передавання: дані, які надходять першими, потрапляють на останній дисплей, а ті, що надходять останніми — на перший. На платі табло реалізоване зниження напруги до рівня, необхідного для коректної роботи РК-дисплеїв (рис. 2.4).



Рисунок 2.4 – Драйвери РК-дисплею

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P2 | Арк. |
|      |      |             |        |      |                   | 44   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

У схемі центральної плати зумер (рис.2.5) підключено через транзистор, база якого керується сигналом з виводу PA5 мікроконтролера. При подачі високого рівня на PA5 транзистор відкривається, тим самим замикаючи один із виводів зумера на загальну мінусову шину. Це викликає звучання зумера.

Керування зумером реалізується шляхом формування імпульсів різної тривалості з використанням затримок . Наприклад, задаючи затримки 150, 100 або 50 мікросекунд, можна формувати сигнали різної довжини, які відтворюються у вигляді коротких, середніх або довгих звукових сигналів. Це дозволяє створювати прості звукові коди для індикації подій.

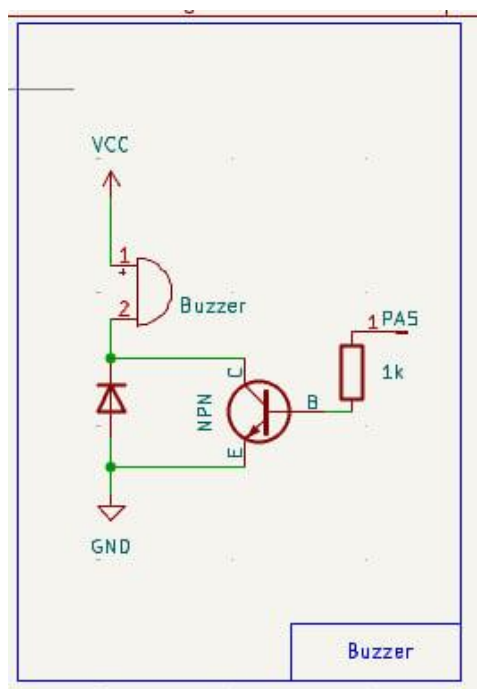


Рисунок 2.5 – Схема управління зумером

У схемотехнічному рішенні центральної плати реалізовано підтримку клавіатури типу 4×4 (рис. 2.6), яка дозволяє опитувати 16 кнопок. Для її підключення використовується два порти мікроконтролера ATmega64: PortF (PF0–PF3) та PortA (PA0–PA3).

Логіка роботи полягає в наступному: виводи PortF0–F3 використовуються як виходи — через них один із рядків клавіатури. Виводи



PortA0–A3 налаштовані як входи і призначені для зчитування стану відповідного стовпця. Таким чином, при натисканні кнопки замикається лінія між активним рядком (PortF) та одним зі стовпців (PortA), і мікроконтролер може визначити координати натиснутої кнопки.

Для забезпечення стабільної логіки зчитування всі входи RA0–RA3 підтягнуті до +5 В через резистори номіналом 2.1 кОм. Це гарантує, що при ненажатій кнопці вхід буде перебувати у стабільному високому рівні, а натискання забезпечить зміну рівня на низький.

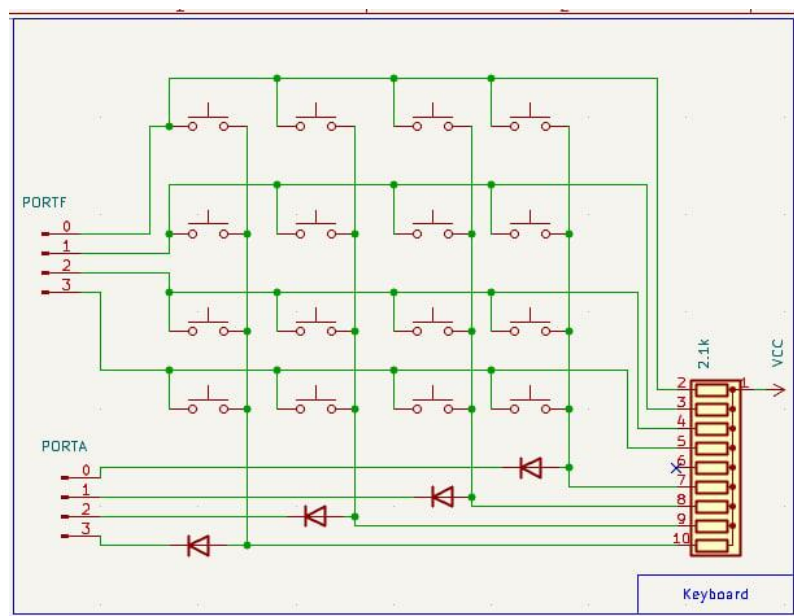


Рисунок 2.6– Схема управління клавіатурою

Датчик пістолету (рис. 2.7) в системі реалізовано за допомогою геркону, розташованого в підпістолетнику. У гарді пістолету вмонтований постійний магніт. Коли пістолет вставляється у підпістолетник, магніт наближається до геркона, і той замикає свої контакти. У даній реалізації один з контактів геркона підключено до виводу PE2 мікроконтролера, а інший — до +5 В. Таким чином, коли геркон замкнутий, на виводі PE2 з'являється високий логічний рівень. У разі, якщо пістолет витягнутий — контакти розімкнені, і PE2 фіксується резистором до землі.

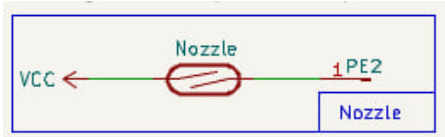


Рисунок 2.7– Схема управління клавіатурою

При обертанні вала дозатора одночасно обертається і вал датчика імпульсів (рис. 2.8), на якому закріплене перфороване диск-коло. Під час обертання отвори диска періодично пропускають світло через оптопару, яка формує цифрові імпульси. Ці імпульси надходять на вхід мікроконтролера PIC16, де відбувається їх попередня обробка.

Далі сформовані імпульси подаються на базу транзистора. У момент відкриття транзистора лінія PD6 мікроконтролера ATmega64 підтягується до землі. У протилежному випадку, коли транзистор закритий, на вході PD6 зберігається високий логічний рівень. Таким чином, ATmega64 приймає чіткі імпульсні сигнали, які використовуються для точного підрахунку об'єму відпущеного пального.

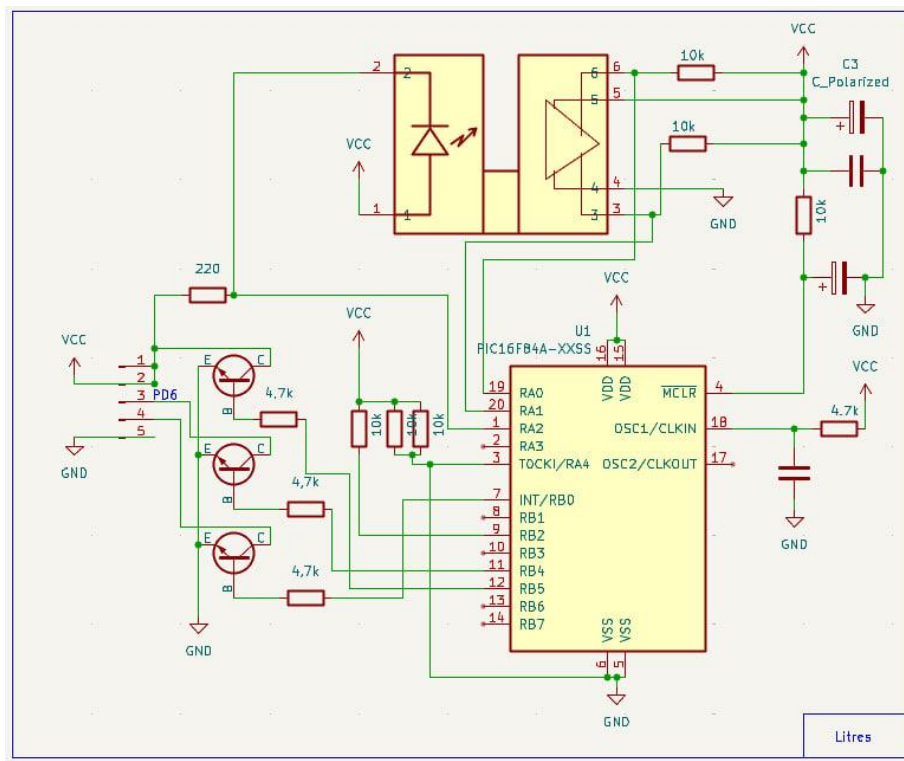


Рисунок 2.8– Схема датчику імпульсу

Фазовий індикатор (рис. 2.9) призначений для перевірки наявності всіх трьох фаз мережі живлення та сигналізації про їх відсутність або неправильну послідовність.

На вході схеми присутні три резистори по 100 кОм, які обмежують струм і подають напругу з кожної фази на відповідні випрямлячі, реалізовані на діодних мостах. Після випрямлення напруга фільтрується електролітичними конденсаторами та подається на оптосимістори. Ці оптосимістори (МОС3063) забезпечують гальванічну розв'язку між високовольною частиною схеми та низьковольною логікою.

У разі наявності всіх фаз, кожен з оптосимісторів буде відкриватися, пропускаючи струм через свій вихід. Ці сигнали обробляються через транзистор і групу світлодіодів з резисторами по 360 Ом, які слугують індикацією. Вивід flag слугує джерелом живлення для оптосимісторів на платі керування двигуном та клапанами, забезпечуючи їхню роботу лише за умови наявності всіх трьох фаз електроживлення.

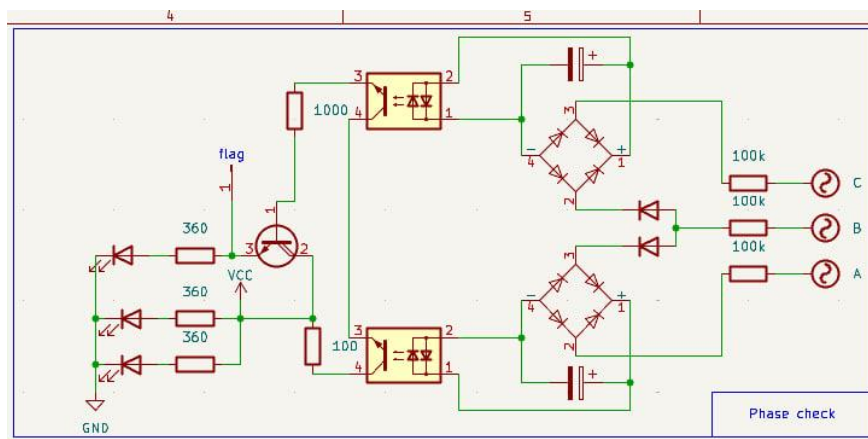


Рисунок 2.9– Схема індикатора фаз

Плата керування двигуном паливного насоса та електромагнітними клапанами реалізує комутацію змінної напруги за допомогою симісторів, які керуються через оптосимістори. Така схема дозволяє забезпечити гальванічну розв'язку між силовими колами та керуючою логікою, що підвищує безпеку роботи системи та зменшує вплив електромагнітних завад.

Керування трифазним двигуном здійснюється через три симістори, підключених до фаз А, В і С. Вони активуються одночасно за допомогою трьох оптосимісторів, на вхід яких подається сигнал з виводу PD7 мікроконтролера. Перед оптосимісторами встановлені резистори номіналом 330 Ом для обмеження струму, а також RC-ланцюги, які зменшують високочастотні завади та захищають силові компоненти.

Керування електромагнітними клапанами виконується аналогічним чином. Кожен канал має свій оптосимістор, який керується з виводів мікроконтролера PG0, PC1. При подачі керуючого сигналу відповідний симістор комутує змінну напругу на котушку клапана, забезпечуючи його відкриття або закриття.

Живлення керуючої частини оптосимісторів подається від виводу flag з плати контролю фаз. Цей сигнал живить оптосимістори, і при його відсутності увесь силовий блок не зможе активуватись, що виконує функцію апаратного блокування при зникненні хоча б однієї з фаз живлення.

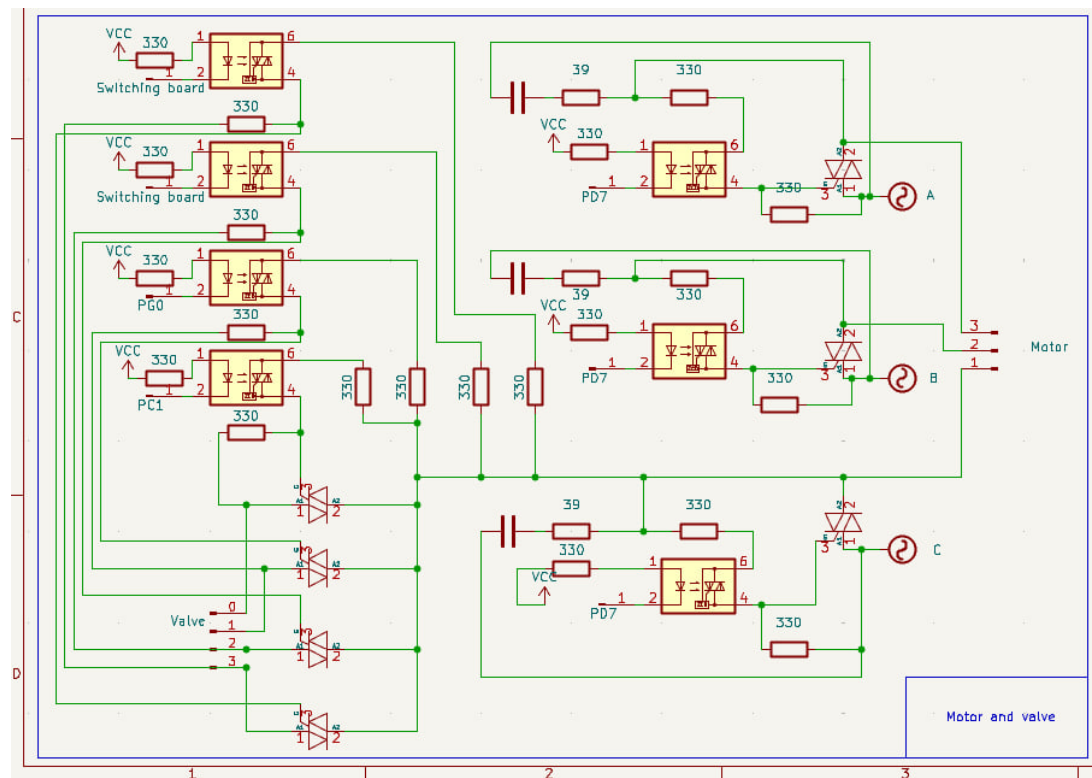


Рисунок 2.10– Схема керування двигуном та клапанами

## 2.2 Програмна структура: модулі, функції та логіка роботи

Для керування індикацією застосовуються драйвери NJU6445F, які передають символи послідовно між собою за допомогою зсувного інтерфейсу. Передача даних до драйверів здійснюється через три виводи мікроконтролера ATmega64: PA6 (DATA) — передача бітів символу, PA7 (CLK) — синхронізація імпульсами, та PG2 (END) — фіксація завершення передачі.

Передача одного символу реалізується через 8 імпульсів CLK (рис. 2.11), під час кожного з яких, одночасно з його фронтом, на лінії DATA встановлюється відповідний біт. Дев'ятий імпульс використовується як службовий, без передачі даних.

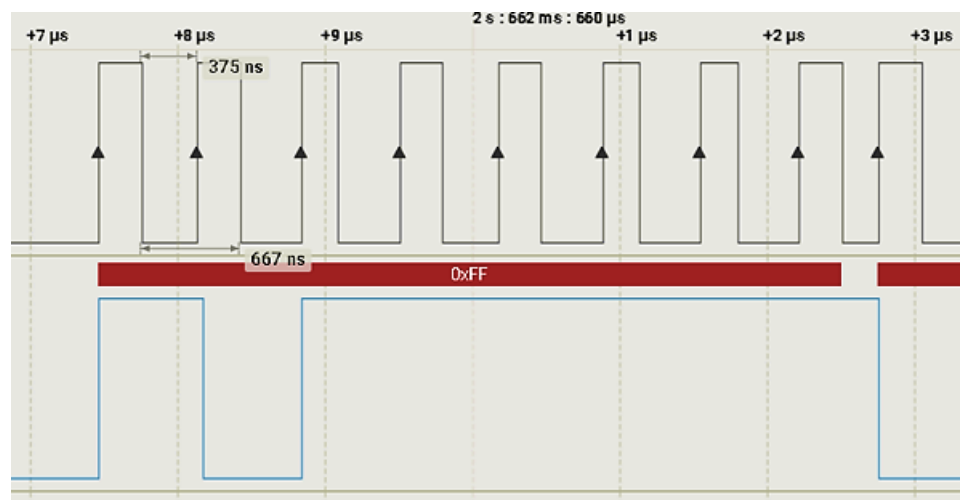


Рисунок 2.11– Один символ

Таким чином, для повного відображення всіх 22 символів (рис. 2.12) на табло необхідно передати 22 пакети по 9 імпульсів, після чого ще додатково три порожні пакети (без активних бітів), щоб завершити цикл передачі. Лише після цього формується імпульс PG2, який фіксує та активує відображення символів на табло.

У програмному коді цей процес реалізований через створення масиву символів у мові C, який потім передається в асемблерну функцію

The timing diagram illustrates the SPI communication between the microcontroller and the ADXL345. The blue signal represents the SPI clock (SCLK), and the red signal represents the SPI data (MOSI). The clock period is 2.417 μs. The data signal shows the sequence of bytes transmitted: 0xF0, 0x5F, 0xA1, 0xD0, 0xE1, 0xF2, 0xF0, 0xF1, 0x7, 0xE0, 0x5F, 0xA1, 0xD1, 0xE1, 0xF3, 0xF0, 0xF1, 0x7, 0xE0, 0x5F, 0xA1, 0xD1, 0xE1, 0xF4, 0xF8, 0x00, 0x00. The diagram also indicates a total time of +0.7 ms and a delay of +0.8 ms.

Після передачі всіх символів функція `sck_3_pak` формує три порожні пакети, а імпульс по PG2 завершує цикл індикації. Якщо потрібно очистити табло тоді просто відправляємо пусті пакети без DATA і тоді індикація очищається .

|             |             |                    |               |             |                   |      |
|-------------|-------------|--------------------|---------------|-------------|-------------------|------|
|             |             |                    |               |             | 171.4321.КР.ПЗ.Р2 | Арк. |
|             |             |                    |               |             |                   | 51   |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ документа</i> | <i>Підпис</i> | <i>Дата</i> |                   |      |

У коді передбачена ініціалізація таймера та глобальне включення переривань, а вся логіка звукових сигналів реалізується у функції обробки переривання ISR(TIMER2\_OVF\_vect) (Додаток А). Для запуску зумера використовується функція set\_buzzer\_mode(Додаток А), яка приймає параметр — режим звучання. Наприклад, режим 1 відповідає короткому сигналу тривалістю 50 мс, режим 2 — двом імпульсам по 150 мс з паузою 50 мс, режим 3 — одному імпульсу 150 мс, а режим 4 — чотирьом коротким імпульсам по 50 мс з паузами по 100 мс (рис.2.13). В залежності від активного режиму, в обробнику переривання відстежується тривалість сигналу та кількість імпульсів за допомогою змінних buz\_counter, buz\_state та pulse\_count.

У разі, якщо обраний режим завершено, таймер вимикається, а пін PA5 очищується, що припиняє звук. Такий підхід дозволяє формувати різні звукові сигнали для різних подій, наприклад: помилок, підтвердження операцій чи попереджень.

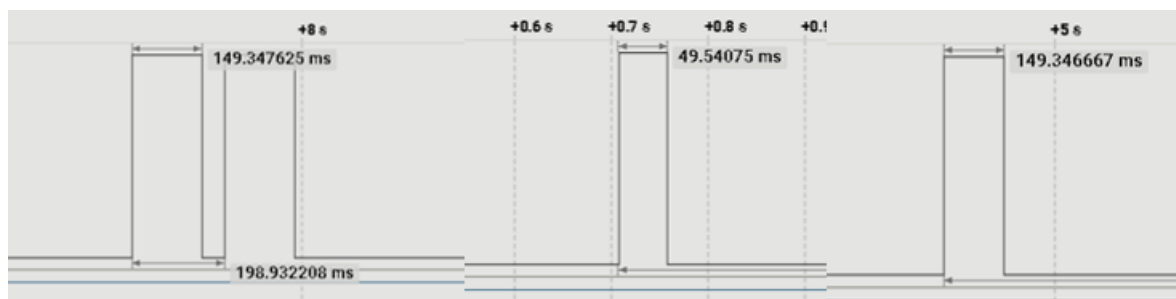


Рисунок 2.13 – Режими зумера

У системі реалізація клавіатури здійснена через матричну схему 4×4 (4 рядки та 4 колонки), де рядки (STR1–STR4) підключені до порту A (PA0–PA3) як входи, а колонки (KOL1–KOL4) — до порту F (PF4–PF7) як виходи. Рядки налаштовані як входи без підтягуючих резисторів, а колонки — як виходи з нульовим початковим станом. Опитування клавіатури відбувається асинхронно за допомогою таймера Timer1 у режимі порівняння.



Основна логіка розміщена в обробнику переривання ISR(TIMER1\_COMPA\_vect) (Додаток А), який викликається з високою частотою. Змінна `pwm_counter` циклічно змінюється від 0 до 78, і на її основі формується сигнал для циклічного активування колонок клавіатури через логічні операції XOR (^). На кожному з етапів (`pwm_counter == 75, 76, 77, 78`) активується одна з колонок, що дозволяє визначити натиснуту клавішу при зчитуванні рівня сигналу на одному з рядків.

Якщо хоча б один рядок знаходиться у високому рівні, встановлюється прапорець `but_c_flag`, що сигналізує про натискання (рис. 2.14). Далі, при певному значенні `pwm_counter` (яке відповідає активній колонці), визначається конкретна натиснута кнопка. Наприклад, при `pwm_counter == 75`, якщо STR1 активний — це кнопка "вверх" (14), а якщо STR4 — "1" (1). Усього підтримується 17 унікальних кнопок, зокрема функціональні (вверх, вниз, вліво, вправо, \*, #) та цифрові (0–9).

Як тільки клавіша виявлена (`pressed_button < 17`), переривання таймера вимикається для запобігання повторного зчитування, що дозволяє обробити натискання лише один раз. Ініціалізація таймера здійснюється у функції `timer1_init_key()` (Додаток А), яка налаштовує таймер у режим СТС з дільником частоти 8 і значенням порівняння `OCR1A = 5`, після чого дозволяє переривання.

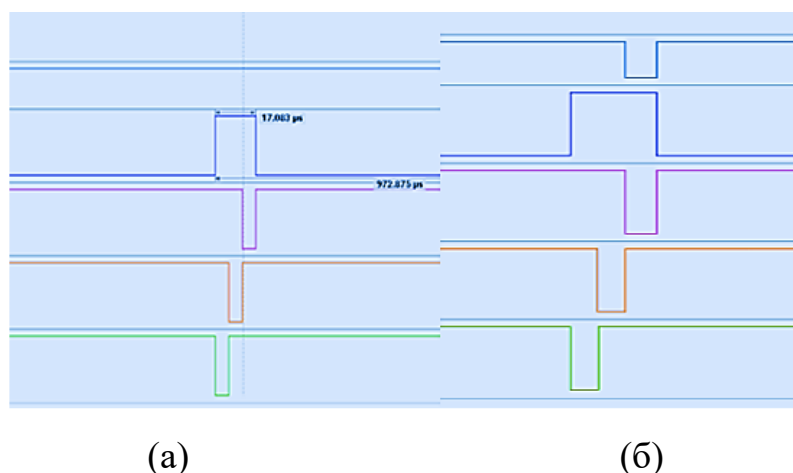


Рисунок 2.14 – (а) Не натиснута кнопка, (б) натиснута

У системі контроль положення паливного пістолета реалізовано за допомогою аналогового компаратора мікроконтролера ATmega64. Для цього використовуються входи PE2 та PE3, які конфігуруються як аналогові (рядок  $\text{DDRE} \&= \sim(1 \ll \text{PE2}) \& \sim(1 \ll \text{PE3});$ ), тобто переведені в режим входу без цифрових функцій.

Ініціалізація компаратора здійснюється у функції AC\_ini()(Додаток А). У регістрі ACSR встановлюється біт ACIE, що дозволяє переривання при будь-якій зміні стану компаратора (перехід з високого в низький або навпаки), а біти ACIS1 і ACIS0 залишаються нулями — це якраз і забезпечує реагування на обидва фронти сигналу. Таким чином, кожна зміна сигналу на входах компаратора викликає обробник ISR(ANALOG\_COMP\_vect) (Додаток А).

У самому обробнику переривання відбувається проста логіка: перевіряється стан біта ACO (Analog Comparator Output), який встановлюється, якщо напруга на позитивному вході компаратора більша, ніж на негативному.

У системі вимірювання об'єму пального датчик імпульсів підключено до виводу PD6 мікроконтролера ATmega64. Цей вивід виконує функцію T1 — лінії лічильного входу для Timer1 у режимі зовнішнього підрахунку імпульсів.

У функції litres\_ini() реалізується ініціалізація цього режиму. Спочатку лінія PD6 налаштовується як вхід ( $\text{DDRD} \&= \sim(1 \ll \text{PD6});$ ), а підтягувальний резистор відключається ( $\text{PORTD} \&= \sim(1 \ll \text{PD6});$ ), оскільки передбачено використання зовнішнього підтягування з боку схеми датчика.

Далі конфігурується таймер 1. Регістр TCCR1A обнуляється, оскільки не використовується PWM або порівняльний режим. В TCCR1B виставляється режим лічильника з зовнішнім джерелом імпульсів — біти  $\text{CS12} = 1, \text{CS11} = 1, \text{CS10} = 0$ , що відповідає "external clock on T1 pin, falling edge" (зовнішній тактовий сигнал на ніжку T1 по спадаючому фронту).

Також через TIMSK відключається генерація переривання по збігу таймера з регістром OCR1A ( $TIMSK \&= \sim(1 \ll OCR1A)$ ), адже таймер використовується лише як лічильник.

У результаті кожен імпульс, сформований з датчика (наприклад, оптопарою з перфорованим диском), буде рахуватись апаратно — без участі процесора

Управління мотором та клапанами у паливороздавальному модулі реалізується шляхом подачі високого логічного рівня на відповідні виводи мікроконтролера ATmega64. Мотор вмикається подачею логічної «1» на вивід PD7, що активує відповідний оптосимістор або реле, через яке здійснюється подача живлення на насос. Клапан малого потоку керується через вивід PG0, а клапан великого потоку — через вивід PC1.

## **2.3 Реалізація ключових алгоритмів керування паливною колонкою**

Після подачі живлення на паливороздавальний модуль, керований мікроконтролером ATmega64, розпочинається процес початкової загрузки портів.

Наступним кроком після виконання початкових налаштувань є тестування блоку індикації. Для цього у коді реалізовано виклик функції `start_paket()` (Додаток А), яка ініціює вивід спеціальної послідовності символів на індикатор для перевірки працездатності кожного сегмента. Така перевірка дозволяє оператору або технічному персоналу візуально переконатися в справності відображення на всіх індикаторах, виявити несправні сегменти або проблеми з підключенням.

На першому етапі функція `start_paket()` (Додаток А) формує масив із 22 однакових символів — це цифра «8» з увімкненою десятковою крапкою (рис. 2.15). Такий символ обрано тому, що він активує всі сегменти семисегментного індикатора одночасно (А, В, С, D, E, F, G, а також DP),

забезпечуючи повний візуальний тест кожного елемента. Цей масив передається на індикатори з певною програмною затримкою, щоб забезпечити візуальне спостереження за результатами.



Рисунок 2.15 – Початкове тестування

Після короткочасного показу «8.» система переходить до другого етапу тестування, де на індикаторах послідовно відображаються цифри від 9 до 0. Ця частина перевірки дає змогу переконатися, що всі цифрові символи відображаються коректно та не спотворені.

Після виконання всіх функцій ініціалізації та попередніх перевірок у систему передаються початкові значення основних параметрів: вартості (cost), об'єму (volume\_now) та ціни за одиницю пального (price). На цьому етапі змінним присвоюються значення за замовчуванням: cost = 0, volume\_now = 0, price = 4232. Значення ціни зберігається у вигляді цілих чисел, де два останні розряди відповідають копійкам, тобто в даному випадку 4232 означає 42 гривні 32 копійки за літр. Ці дані передаються у функцію parse\_number\_to\_digits()(Додаток А), яка виконує перетворення числових значень у масив символів для подальшого виводу на індикатор. Отриманий масив символів number формується у форматі, придатному для відображення вартості, об'єму та ціни. Далі за допомогою функції process\_digits1() здійснюється фактична передача масиву символів до блоку індикації. Далі активується звукова індикація — викликається функція set\_buzzer\_mode(3),

яка запускає короткий сигнал бузера, що вказує на завершення ініціалізації та готовність пристрою до подальшої роботи.

Після проходження етапу тестування індикації та відображення версії прошивки система переходить у режим очікування початкових дій користувача — так зване передналивальне налаштування. На цьому етапі модуль готовий до взаємодії з оператором, і подальші дії визначаються положенням перемикача «Меню», який є своєрідним тумблером і виступає єдиним способом вибору між стандартним режимом заправки та режимом налаштувань.

Якщо тумблер переведено у положення «Меню», користувач отримує доступ до меню налаштувань. У цьому режимі на індикаторі послідовно виводяться пункти меню, що дають змогу переглядати або змінювати внутрішні параметри системи: ціну, сервісні значення, параметри контролю налива тощо. Усі зміни в меню виконуються виключно через клавіатуру, яка є основним інтерфейсом введення. За допомогою клавіш користувач може переходити між пунктами меню, змінювати значення параметрів та підтверджувати введені дані (рис. 2.16).



Рисунок 2.16-Зміна ціни за літр

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P2 | Арк. |
|      |      |             |        |      |                   | 57   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

У разі, якщо тумблер знаходиться не в положенні «Меню», система автоматично переходить у робочий режим, призначений для передналивального налаштування. У цьому режимі оператор має можливість задати актуальну ціну за літр, а також вибрати або суму, на яку потрібно заправити паливо, або обсяг пального в літрах(рис.2.17). Усі ці дії також виконуються через клавіатуру. Користувач за допомогою відповідних клавіш змінює значення параметрів, а система оперативно відображає зміни на індикаторі.



Рисунок 2.17-Введення літрів заправки

Після завершення введення необхідних значень модуль переходить у стан готовності до початку заправки. У цей момент можливе увімкнення клапана налива, і система чекає на запуск фактичного процесу відпуску пального.

Наступним ключовим етапом роботи системи є процес заправки, реалізований у функції `refueling()`(Додаток А). Саме ця функція відповідає за

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P2 | Арк. |
|      |      |             |        |      |                   | 58   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

контроль і виконання операції налива пального згідно з заданими параметрами — сумою або літражем.

На початку роботи функції на індикатор виводиться цифра "8", що використовується як умовний маркер початку операції. Виведення символу здійснюється шляхом виклику функції `send_digit22(8)`, яка через асемблерні процедури передає дані на індикатор. Після цього вмикається звукова індикація — короткий сигнал, що формується функцією `set_buzzer_mode(3)`, і затримка, яка забезпечує чітке сприйняття етапу переходу в режим заправки.

Далі відбувається ініціалізація початкових значень для обліку об'єму пального. Викликається функція `litres_ini()`, яка, як правило, відповідає за попереднє налаштування датчика імпульсів або лічильника, що фіксує кількість налитого пального. Значення таймера `TCNT1` скидається, а також обнуляється змінна `timer1`, яка надалі зберігатиме кількість імпульсів. Значення змінної `cost` також обнуляється, оскільки процес заправки розпочинається з нульової вартості. Обсяг, який було введено на передньому етапі, зберігається в змінній `volume_old`, а поточне значення наливу `volume_now` скидається до нуля.

Після цього викликається функція `parse_number_to_digits()`, яка перетворює поточні значення вартості (`cost`), налитого об'єму (`volume_now`) та ціни (`price`) у формат, що підходить для відображення на індикаторі. Отриманий масив передається у функцію `process_digits1()`, яка, використовуючи асемблерні підпрограми, формує відповідну послідовність імпульсів для індикаторного блоку.

На цьому етапі система фізично активує виконавчі елементи. На порті `PORTD` вмикається сигнал для активації мотору (`MOT`), на `PORTC` подається сигнал на відкриття клапану великого потоку (`VCUT`), а також вмикається клапан малого потоку через інверсну логіку (`VLOW` на `PORTG` встановлюється в 0) (рис. 2.18).

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.Р2 | Арк. |
|      |      |             |        |      |                   | 59   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |



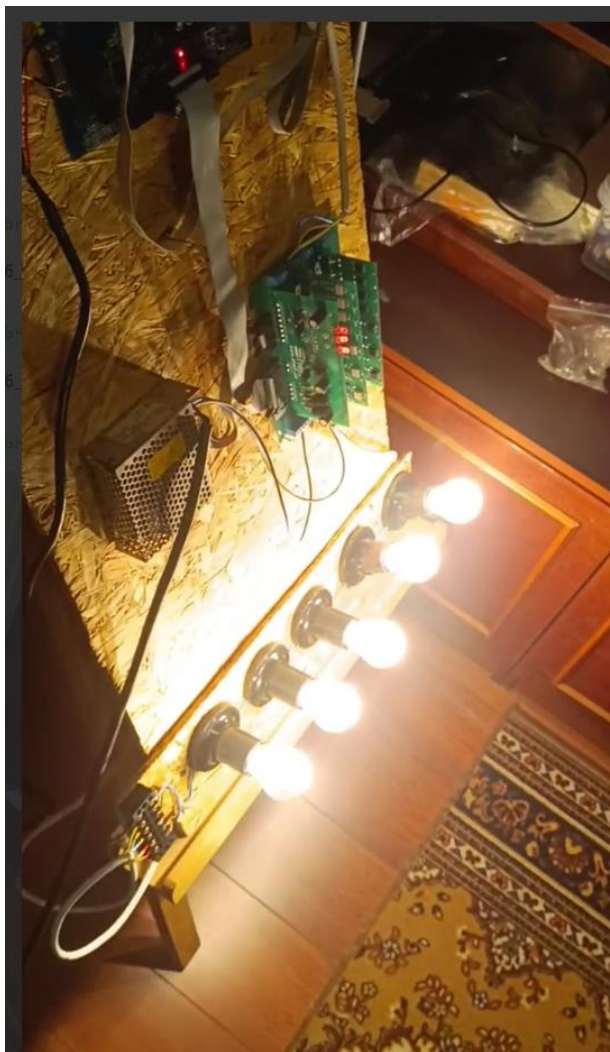


Рисунок 2.18-Запуск мотору та клапанів

Основний цикл налива реалізується в циклі `while`, який триває доти, доки значення `volume_now` не досягне значення `volume_old`, тобто поки не буде наливо заданий об'єм пального (рис. 2.19). У тілі циклу значення таймера `TCNT1` зчитується в змінну `timer1`, яка використовується як лічильник імпульсів. Поточний налитий об'єм присвоюється змінній `volume_now`, і на його основі обчислюється поточна вартість за формулою:  $cost = (volume\_now * price) / 100$ , де враховується масштабування для переведення одиниць.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р2 | Арк. |
|      |      |             |        |      |                   | 60   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



Рисунок 2.19-Зміна літражу та суми при обертанні датчику іпульсу

Якщо система виявляє, що залишилося мало до досягнення цільового об'єму ( $\text{volume\_old} - \text{volume\_now} < \text{SLOWDOWN\_THRESHOLD}$ ), то вона автоматично перемикається на малий потік: закривається клапан великого потоку (VCUT) і активується клапан малого потоку (VLOW), що дозволяє уникнути переливу та забезпечити точність відпуску.

Після кожного циклу обчислення нових значень виконується оновлення індикації через функцію `parse_number_to_digits()` та `process_digits1()`.

У фіналі, коли заданий об'єм досягнуто або перевищено, виконавчі елементи деактивуються: мотор вимикається, обидва клапани перекриваються. Це означає завершення процесу заправки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P2 | Арк. |
|      |      |             |        |      |                   | 61   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

## Висновки до розділу 2

У цьому розділі було детально розглянуто процес розробки програмного забезпечення для мікроконтролера, що виконує керування центральною платою паливороздавального комплексу (ПРК). На основі обраної апаратної платформи — мікроконтролера ATmega64 — були обґрунтовані основні засоби розробки, зокрема середовище програмування, мова C та використання асемблерних вставок для роботи з індикацією. Це дозволило ефективно використовувати ресурси мікроконтролера та забезпечити надійну й швидкодіючу роботу системи.

Особливу увагу було приділено схемотехнічним рішенням центральної плати, розробленої виробником. Програмна реалізація враховує всі особливості апаратної частини: інтерфейси для зв'язку з індикацією, клапанами, мотором, датчиками імпульсів, а також вузол клавіатури. Програмний код був адаптований під існуючу структуру сигналів, логіку інверсій, електричні особливості входів/виходів.

У підрозділі, присвяченому програмній структурі, описано модульний підхід до побудови прошивки: виокремлено логіку індикації, зчитування клавіатури, обробку станів системи, звукову сигналізацію та інші функціональні частини. Це дозволяє легко масштабувати або змінювати систему відповідно до майбутніх потреб.

Окремо реалізовано ключові алгоритми, які забезпечують стабільну роботу ПРК — зокрема, початкову ініціалізацію системи, передналивальні налаштування, перевірку елементів індикації, вибір режиму заправки, налаштування ціни, обсягу або суми, а також сам процес заправки з плавним перемиканням потоку. Усі ці алгоритми протестовані та синхронізовані з апаратними подіями, що гарантує коректну роботу системи в реальних умовах експлуатації.

Таким чином, створене програмне забезпечення забезпечує повноцінне функціонування центральної плати ПРК відповідно до заданої логіки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р2 | Арк. |
|      |      |             |        |      |                   | 62   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

|             |      |                  |                                                                                     |      |                                           |                                |      |        |
|-------------|------|------------------|-------------------------------------------------------------------------------------|------|-------------------------------------------|--------------------------------|------|--------|
|             |      |                  |                                                                                     |      | 171.4321.КР.ПЗ.РЗ                         |                                |      |        |
|             |      |                  |                                                                                     |      |                                           |                                |      |        |
| Змн.        | Арк. | № документа      | Підпис                                                                              | Дата |                                           |                                |      |        |
| Розробник   |      | Піряник А.Р.     |  |      | Тестування та перевірка<br>працездатності | Літ.                           | Арк. | Аркуші |
| Консультант |      |                  |                                                                                     |      |                                           |                                | 63   |        |
| Н. контр.   |      | Добрінова Л. П.  |  |      |                                           | НУК<br>імені адмірала Макарова |      |        |
| Керівник    |      | Жук О. К.        |  |      |                                           |                                |      |        |
| Зав. каф.   |      | Рябенський В. М. |  |      |                                           |                                |      |        |

## РОЗДІЛ 3. ТЕСТУВАННЯ ТА ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ

### 3.1. Методика тестування

Для перевірки правильності роботи програмного забезпечення та апаратної частини паливороздавального модуля була розроблена методика тестування, що охоплює всі ключові вузли системи: індикацію, клавіатуру, модулі керування та датчики.

Одним із основних інструментів, використаних у процесі налагодження, став логічний аналізатор (рис. 3.1), який застосовувався для дослідження цифрових сигналів на виходах мікроконтролера ATmega64.



Рисунок 3.1 – Логічний аналізатор на 8 каналів[13]

На початковому етапі він використовувався для дослідження оригінальної прошивки, встановленої на контролер виробника. За допомогою аналізатора вдалося зчитати та проаналізувати сигнали, які передавалися мікроконтролером до різних периферійних пристроїв, зокрема до РК-дисплея (рис. 3.2), клавіатури, моторів та електроклапанів.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 64   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

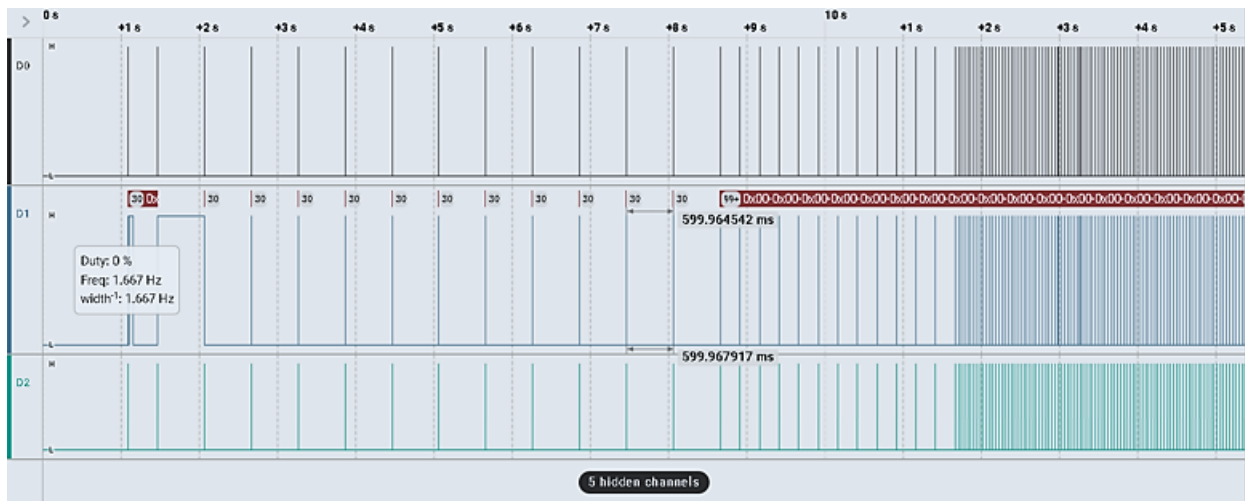


Рисунок 3.2 – Пакети для РК-дисплею

Це дозволило зрозуміти логіку управління, структуру переданих даних (рис. 3.3), порядок тактування, а також визначити часові характеристики імпульсів. Отримані результати допомогли правильно реалізувати передачу даних у власному коді, зокрема забезпечити коректну роботу функцій індикації, обробки натискань клавіш та керування звуковою сигналізацією. Перед підключенням модулів до стенду нова прошивка перевірялась через логічний аналізатор на відповідність параметрів сигналів – це дозволяло виявити помилки на ранньому етапі, уникнути фізичного пошкодження компонентів та забезпечити правильну взаємодію мікроконтролера з апаратною частиною.

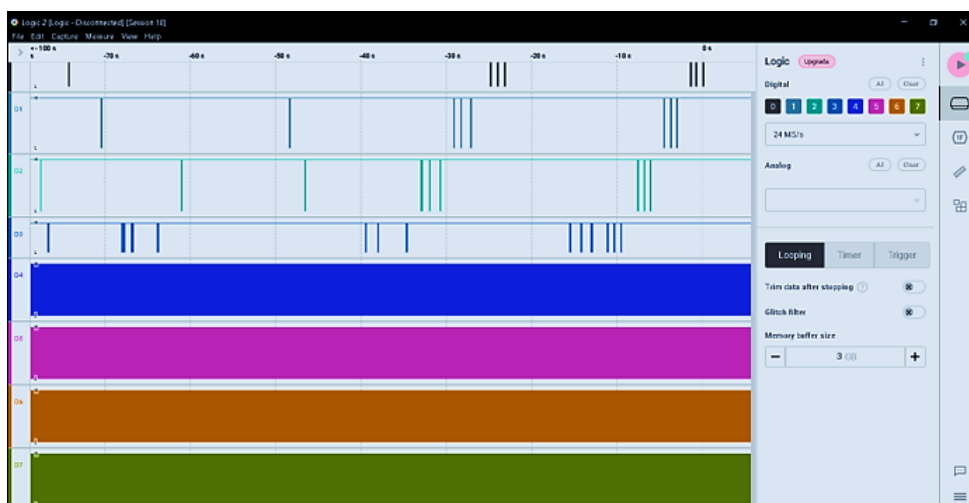


Рисунок 3.3 – Спрацювання клавіатури



Таким чином, логічний аналізатор став не лише діагностичним засобом, а й важливим інструментом для зворотного інжинірингу та верифікації коректності реалізованих алгоритмів у прошивці.

Для перевірки алгоритмів заправки, налаштувань та інших дій, які виконуються за участі користувача, було створено стенд для тестування (рис. 3.4).

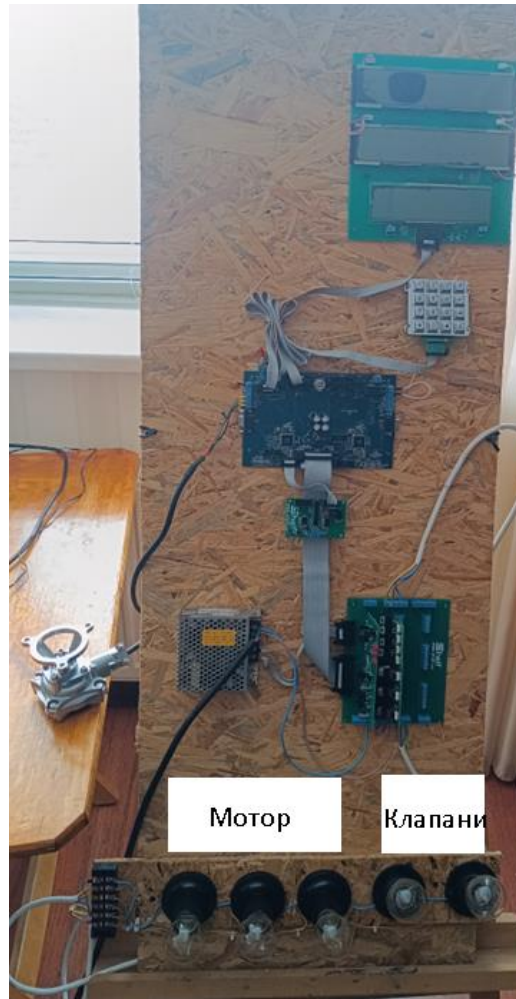


Рисунок 3.4 – Стенд

Стенд відіграв важливу роль на всіх етапах розробки прошивки, особливо на початковому, коли виникала необхідність протестувати відремонтовані або нові апаратні модулі до встановлення їх у повноцінну паливороздавальну колонку. Завдяки наявності стенду стало можливим швидко перевірити працездатність окремих вузлів: індикаторів, клавіатури, датчиків, силових елементів, клапанів та реле. Це дозволило оперативно



виявляти дефекти, а також забезпечити безпечну роботу з електронними компонентами в контрольованому середовищі.

На наступному етапі стенд активно використовувався під час написання та модульного тестування програмного забезпечення. Він допоміг виявити типові проблеми, які виникають при взаємодії окремих програмних модулів, таких як неправильне оновлення індикації, затримки в обробці клавіатури, збої в логіці перемикання клапанів або некоректне формування даних для відображення. Завдяки можливості фізичного спостереження за результатами виконання коду та швидкому циклу «зміна коду – тестування», стенд дозволив суттєво пришвидшити процес розробки та забезпечити високу надійність і стабільність роботи прошивки в реальних умовах експлуатації.

### 3.2. Аналіз помилок та їх усунення

Аналіз помилок під час роботи з індикацією виявив низку суттєвих проблем, пов'язаних із недостатньою швидкістю програмного коду, написаного мовою C. Через використання кварцового генератора на 7,3728 МГц і високі вимоги до точності та швидкості передачі сигналів на індикатор, стандартні програмні засоби на мові C не забезпечували необхідного рівня синхронізації. В результаті оновлення дисплея відбувалося із затримками, інформація відображалася з помилками або зовсім не з'являлася.

Функціонал оригінальної прошивки, яка була більш оптимізованою, неможливо було відтворити лише за допомогою C-коду — потрібна була висока точність таймінгів і мінімальні затримки між імпульсами сигналів. Тому було прийнято рішення реалізувати критичну частину коду, що відповідає за вивід інформації на індикатор, на мові асемблера. Це дозволило забезпечити точний контроль над сигналами (DATA, CLK) та забезпечити стабільну роботу індикації з мінімально можливою затримкою.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 67   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

Крім того, на початковому етапі виникали проблеми з відображенням самих символів. Було складно визначити, який сегмент дисплея відповідає за який біт керування. Тому символи, виведені на індикатор, часто мали неправильну форму або були частково неактивні (рис. 3.5). Лише після детального аналізу та тестування вдалося встановити точну відповідність між бітом даних і конкретним сегментом, що забезпечило коректне відображення як цифр, так і букв.



Рисунок 3.5 – Не коректне відображення цифри

Завдяки цим заходам вдалося досягти максимальної сумісності з апаратною логікою індикатора та реалізувати швидкий, надійний механізм виводу інформації, що повністю відтворює логіку оригінальної системи.

У процесі реалізації клавіатурного інтерфейсу було виявлено низку проблем, які суттєво впливали на зручність користування системою. Основною проблемою став дребезг контактів при натисканні кнопок, що спричиняло фіксацію кількох натискань замість одного або ж повну відсутність реакції системи. Також у первинному варіанті обробка натискань виконувалася в основному циклі програми, що призводило до затримок у реакції інтерфейсу, особливо під час одночасного виконання інших задач.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 68   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

Для вирішення цих проблем було реалізовано обробку клавіатури через таймерне переривання ISR(TIMER1\_COMPA\_vect). Такий підхід дозволив стабільно опитувати клавіатуру з фіксованою періодичністю, забезпечуючи фільтрацію дребзгу контактів і високу чутливість до натискань.

У середині процедури переривання реалізовано механізм поступового включення і вимкнення ліній колонок клавіатурної матриці через регістр PORTF. На кожному з етапів (pwm\_counter від 75 до 78) активується одна з колонок, після чого опитуються чотири стрічкові лінії (рядки), підключені до порту PINA. Якщо на якомусь рядку фіксується низький рівень, система визначає, яка саме кнопка натиснута, і присвоює відповідне значення змінній pressed\_button.

Для уникнення повторного спрацювання за один цикл натискання, використовується прапор but\_c\_flag, який блокує повторну обробку до завершення циклу. Окрім того, після зчитування натискання система тимчасово вимикає переривання по TIMSK &= ~(1 << OCIE1A);, щоб уникнути надмірного навантаження на процесор і дублювання подій.

Завдяки цьому рішенню вдалося повністю усунути проблему дребзгу та зробити інтерфейс клавіатури чітким, швидким і надійним.

У процесі інтеграції апаратних модулів виникли труднощі з коректною обробкою сигналів від двох важливих компонентів: датчика пістолета та датчика імпульсів. На початкових етапах система некоректно визначала момент зняття або повернення пістолета на місце(рис.3.6). Сигнал від датчика або надходив із запізненням, або фіксувався кілька разів підряд, що спричиняло збої в логіці процесу заправки. Було встановлено, що для цього датчика в оригінальній конструкції використовується аналоговий компаратор мікроконтролера. Для обробки цього сигналу було реалізовано переривання ISR(ANALOG\_COMP\_vect), яке дозволяє миттєво реагувати на зміну логічного стану. Завдяки цьому вдалося значно зменшити затримку реакції системи та забезпечити стабільне визначення положення пістолета. Після

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 69   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

реалізації цього підходу система почала коректно визначати стан пістолета: піднятий він чи ні.

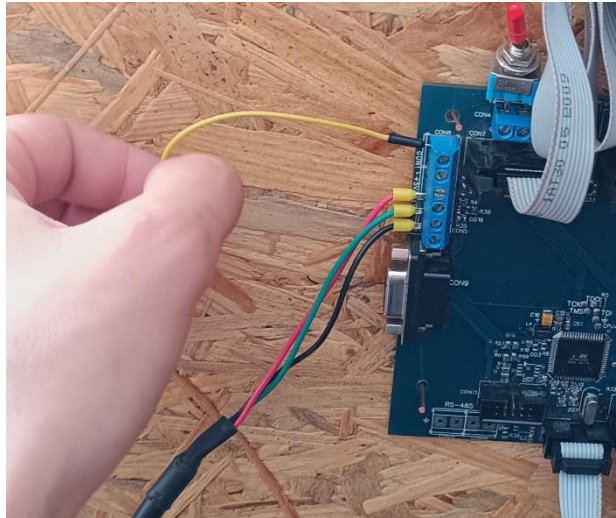


Рисунок 3.6 – Умовний датчик пістолету

Ще однією важливою складовою була система обліку кількості виданого пального, яка реалізується через імпульсний датчик, підключений до пін T1 (PD6) мікроконтролера. На початку були випадки пропуску імпульсів або, навпаки, їх подвоєння. Такі збої могли призвести до помилкового обчислення об'єму пального, що критично для точності роботи паливороздавального комплексу. Щоб забезпечити стабільний облік імпульсів, було налаштовано лічильник Timer1 у режимі зчитування імпульсів із зовнішнього джерела. Такий режим дозволив апаратно рахувати кожен імпульс, що надходив від датчика, забезпечуючи високу точність і стабільність навіть при високій частоті імпульсів (рис. 3.7). Також був усунутий ризик втрати імпульсів, який виникав при спробі обробки цих сигналів у програмному циклі. Завдяки правильному використанню апаратних ресурсів мікроконтролера вдалося усунути критичні помилки, пов'язані з обробкою датчиків, та забезпечити стабільну, точну і своєчасну роботу системи контролю пального.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 70   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |



Рисунок 3.7 – Імітування оберту дозатора

Під час розробки та тестування керування силовим блоком виникли кілька суттєвих проблем, які впливали на стабільність та точність роботи всієї системи (рис. 3.8). Найбільш критичною була ситуація, коли одразу після подачі живлення відбувалося мимовільне увімкнення мотору або електромагнітних клапанів. Це було пов'язано з тим, що силові ключі активуються низьким рівнем сигналу, а порти мікроконтролера після скидання перебувають у стані з високим імпедансом. У такому стані керуючі виводи могли сприйматись як логічний нуль, що спричиняло небажане ввімкнення виконавчих елементів ще до старту основної програми.



Рисунок 3.8 –Ввімкнутий мотор та клапани

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 71   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



Для усунення цієї проблеми в кодї була реалізована ініціалізація всіх керуючих виводів у високий рівень одразу після запуску. Це забезпечує вимкнений стан силових елементів, таких як реле, транзистори та інші ключі, і дозволяє системі стартувати у безпечному режимі без ризику випадкового включення мотору чи клапанів.

Ще одна важлива проблема стосувалася послідовності відключення клапанів великого та малого потоку (рис. 3.9). У початковій реалізації обидва клапани вимикалися одночасно або навіть у неправильному порядку, що призводило до неточностей у фінальному об'ємі налитого пального. Для точного дозування пального необхідно, щоб після вимкнення клапана великого потоку певний короткий час ще працював клапан малого потоку — це дозволяє завершити заправку з високою точністю до встановленого об'єму.



Рисунок 3.9 – Ввимкнутий клапан великого потоку

Після аналізу поведінки оригінального ПЗ та експериментального тестування на стенді, було впроваджено послідовне відключення: спочатку вимикається клапан великого потоку, а через контрольовану затримку — клапан малого потоку. Це дозволило значно покращити точність наливу та уникнути гідроударів у системі, що особливо важливо для довговічності механічних елементів колонки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 72   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

### 3.3. Результати випробувань

Після завершення основного етапу розробки програмного забезпечення було проведено низку випробувань для перевірки коректної роботи всіх функціональних блоків системи. Тестування проводилось як на лабораторному стенді, так і на реальній апаратній частині паливороздавальної колонки.

Одним із ключових показників була стабільна робота індикації. Завдяки переходу на асемблерну реалізацію функцій передачі символів вдалося забезпечити своєчасне та безперебійне оновлення відображення, навіть при високому навантаженні на мікроконтролер. Всі символи виводились коректно, без збоїв, миготіння або артефактів.

Клавіатура після впровадження обробки через переривання показала хорошу реакцію на натискання — спостерігалася чітка робота без затримок, зникли помилкові спрацювання та ефект «дребезгу» контактів. Усі кнопки розпізнавалися точно, навіть при швидкому наборі команд.

Датчики стану пістолета та імпульсного підрахунку працювали стабільно після налаштування апаратних модулів контролера. Випробування показали, що стан пістолета визначається своєчасно, без затримок, а імпульсні сигнали чітко підраховуються без пропусків, що важливо для точного обліку об'єму налитого пального.

Силовий блок після усунення початкових проблем продемонстрував передбачувану поведінку: мотор запускається лише після чіткого дозволу від мікроконтролера, а клапани спрацьовують у правильній послідовності. Особливо важливим було підтвердження точності вимкнення клапанів — навіть при встановлених малих об'ємах наливу вдалося досягти похибки в межах допустимих значень.

У результаті випробувань було підтверджено, що система працює стабільно.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 73   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |



### 3.4. Перевірка відповідності технічним вимогам

Після завершення розробки програмного забезпечення та проведення серії випробувань було здійснено перевірку відповідності розробленого модуля керування паливороздавальною колонкою технічним вимогам. Оцінка охоплювала як функціональні можливості, так і відповідність часовим, електричним та інтерфейсним параметрам. При проектуванні програмного коду особливу увагу приділялася тому, щоб максимально наблизитися до оригінальної реалізації виробника — як з точки зору логіки роботи, так і з погляду взаємодії окремих модулів. Це дозволило забезпечити сумісність із наявною апаратною частиною та спростити подальше тестування.

Система успішно відображає необхідну інформацію на семисегментному індикаторі — ціна за літр, кількість пального, підсумкова сума та службові повідомлення виводяться чітко та синхронно. Завдяки оптимізації коду передачі символів на асемблері та використанню кварцу на 7.3728 МГц вдалося забезпечити стабільну роботу індикації без візуальних артефактів (рис. 3.10).



Рисунок 3.10 – Правильне відображення

Клавіатурний ввід реалізовано з використанням переривання таймера, що дозволило позбутися ефекту дребезгу контактів та забезпечити миттєве

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 74   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

реагування на натискання кнопок. Підтверджено коректне розпізнавання цифрових значень, команд меню, переходів та підтвердження/скасування.

У системі реалізовано три основні режими наливу: по літрах, по сумі та безлімітний. У першому випадку користувач вводить бажаний об'єм, і система автоматично обчислює суму. У другому — навпаки: вводиться сума, і контролер розраховує обсяг пального. Третій режим передбачає налив без обмеження — для сервісних або спеціальних випадків. Кожен з цих режимів був протестований на функціональному стенді, підтверджено їх повну працездатність.

Важливою частиною є базове меню налаштувань, у якому можна встановити ціну за літр, задати пароль доступу до сервісу, обнулити лічильники або змінити конфігурацію силового блоку. Навігація по меню інтуїтивно зрозуміла, реалізована через клавіші управління і працює без затримок (рис. 3.11).



Рисунок 3.11 – Перша вкладка меню «зміна ціни»

Система також реалізує контроль пістолета за допомогою аналогового компаратора. Реакція контролера на підняття або опускання пістолета є миттєвою, що дозволяє точно зафіксувати початок і завершення наливу. Для обліку кількості пального використовується імпульсний витратомір,

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 75   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

підключений через вхід Т1. Зовнішній режим таймера 1 забезпечує стабільне і точне рахування імпульсів навіть на високих частотах.

Контроль наливу реалізований через цілу низку механізмів: перевірка кількості імпульсів у реальному часі, автоматичне вимкнення клапана великого потоку перед малим для забезпечення точної дози.

Керування силовими елементами — мотором і двома клапанами — реалізовано з урахуванням технічних вимог (рис. 3.12). Початкові рівні на керуючих пінових були налаштовані на високий рівень, що виключає самовільне ввімкнення при старті системи. Вимкнення клапанів відбувається в заданій послідовності: спочатку великого, потім малого потоку, що забезпечує точність наливу та запобігає перевищенню об'єму.

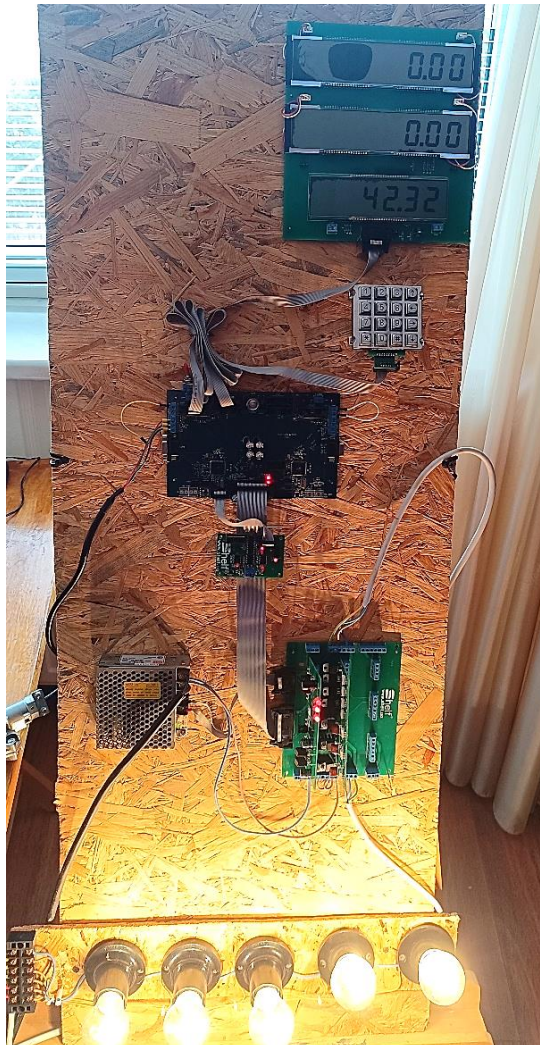


Рисунок 3.12 – Початок заправки

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
| Змн. | Арк. | № документа | Підпис | Дата |                   | 76   |

### Висновки до розділу 3

У результаті проведення повного циклу тестування програмного забезпечення та апаратної частини було підтверджено працездатність системи керування паливороздавальною колонкою відповідно до заданих технічних вимог. Реалізована методика випробувань дозволила не лише перевірити функціональність усіх основних модулів, а й виявити та усунути низку критичних і другорядних помилок, що виникали на етапі інтеграції.

Зокрема, завдяки поступовому аналізу поведінки індикаторів, клавіатури, імпульсних і аналогових датчиків, силових елементів, а також логіки режимів наливу, вдалося досягти стабільної та узгодженої роботи системи. Помилки, пов'язані з некоректним відображенням, затримками у керуванні чи помилковими спрацюваннями, були виправлені шляхом оптимізації коду, використання апаратних переривань, правильного налаштування таймерів і стартових рівнів.

Кінцеві результати випробувань засвідчили, що система повністю відповідає функціональним, часовим, інтерфейсним та електричним параметрам, а її структура дозволяє гнучко адаптувати логіку до потреб користувача або особливостей конструкції. Реалізація основних режимів наливу, базового меню налаштувань, а також контролю за всіма ключовими подіями гарантує як точність дозування пального, так і безпечну експлуатацію. Таким чином, розроблений програмно-апаратний комплекс є готовим до подальшого впровадження та використання в складі паливороздавального обладнання.

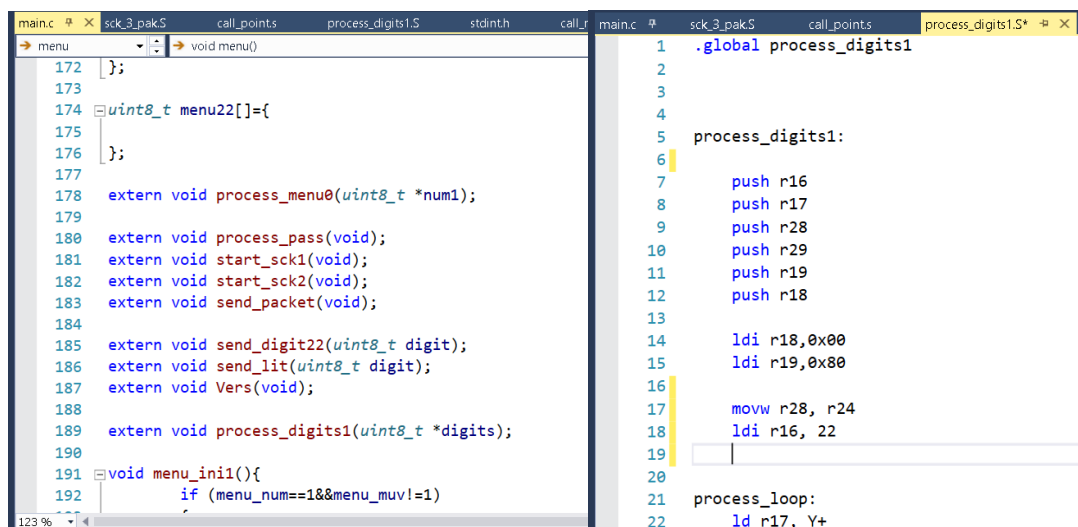
|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.РЗ | Арк. |
|      |      |             |        |      |                   | 77   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

|             |      |                  |                                                                                     |      |                                                                  |                                |      |        |
|-------------|------|------------------|-------------------------------------------------------------------------------------|------|------------------------------------------------------------------|--------------------------------|------|--------|
|             |      |                  |                                                                                     |      | 171.4321.КР.ПЗ.Р4                                                |                                |      |        |
| Змн.        | Арк. | № документа      | Підпис                                                                              | Дата | Практичне значення виконаної роботи та шляхи подальшого розвитку | Літ.                           | Арк. | Аркуші |
| Розробник   |      | Піряник А.Р.     |  |      |                                                                  |                                |      |        |
| Консультант |      |                  |                                                                                     |      |                                                                  |                                | 78   |        |
| Н. контр.   |      | Добрінова Л. П.  |  |      |                                                                  | НУК<br>імені адмірала Макарова |      |        |
| Керівник    |      | Жук О. К.        |  |      |                                                                  |                                |      |        |
| Зав. каф.   |      | Рябенський В. М. |  |      |                                                                  |                                |      |        |

## РОЗДІЛ 4. ПРАКТИЧНЕ ЗНАЧЕННЯ ВИКОНАНОЇ РОБОТИ ТА ШЛЯХИ ПОДАЛЬШОГО РОЗВИТКУ

### 4.1. Отримані навички в програмуванні мікроконтролерів

У процесі створення програмного забезпечення для паливороздавального модуля на базі мікроконтролера ATmega64 мною були набуті важливі практичні навички як у низькорівневому програмуванні, так і в структурній організації коду за принципами модульного підходу. Одним із ключових досягнень стало опанування написання асемблерних функцій для оптимізації критичних ділянок, зокрема для швидкої передачі даних до індикатора. Завдяки цьому вдалося значно підвищити продуктивність порівняно з аналогічною реалізацією на мові Сі. Я навчився викликати асемблерні функції з мови Сі (рис. 4.1), передавати їм параметри та забезпечувати їхню коректну взаємодію з іншими частинами програми.



(а)

(б)

Рисунок 4.1— Ініціалізація зовнішніх асемблерних функцій (а) ,приклад асемблерної функції (б)

Також було здобуто практичний досвід роботи з апаратними модулями мікроконтролера. Зокрема, використання аналогового компаратора дало можливість точно фіксувати стан пістолета без затримок. Робота з таймером

у режимі зовнішнього лічильника дозволила реалізувати точний облік імпульсів з витратоміра. За допомогою переривань таймера реалізовано стійке зчитування клавіатури без дребезгу контактів. Я навчився правильно налаштовувати порти вводу/виводу для керування силовими елементами, враховуючи необхідність встановлення відповідного початкового логічного рівня для запобігання помилковому спрацюванню при подачі живлення.

Окрему увагу під час розробки приділено модульній організації програмного коду. Весь проєкт структуровано у вигляді окремих логічних модулів — індикація, клавіатура, облік, меню, керування тощо. Це забезпечило зручність супроводу, тестування, повторного використання коду та його масштабування. Я навчився створювати заголовкові файли для опису інтерфейсів між модулями, чітко відокремлювати апаратнозалежну логіку від функціональної, а також забезпечувати стабільну взаємодію між окремими частинами програми.

У результаті виконаної роботи я здобув комплексні навички: написання асемблерного коду і його інтеграція в проєкт на мові Cі, робота з апаратними ресурсами мікроконтролера, побудова стабільних алгоритмів з використанням переривань, реалізація модульної структури та з'єднання окремих модулів у єдину функціональну систему. Ці знання стали міцною основою для подальшої професійної діяльності в галузі розробки мікроконтролерних систем та автоматизації.

#### **4.2. Формування практичних навичок діагностики та ремонту ПРК**

У процесі виконання дипломного проєкту було сформовано широкий спектр практичних навичок, пов'язаних із діагностикою, ремонтом та налагодженням паливороздавальних колонок, зокрема обладнання виробництва компанії «Шельф». Велику роль у цьому відіграв спеціалізований стенд, який дозволив досліджувати роботу окремих модулів,

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 80   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |



імітувати реальні ситуації та проводити комплексну перевірку функціональності системи.

За допомогою стенду проводилася діагностика різних апаратних компонентів: силових модулів, що керують клапанами і мотором; датчиків — імпульсу та пістолету; клавіатурного модуля, РК-дисплея, блоків живлення та інших елементів. Це дозволило здобути навички локалізації несправностей і визначення джерела проблем — як у програмному забезпеченні мікроконтролера, так і в апаратній частині системи.

Важливу роль у процесі діагностики відіграв логічний аналізатор — інструмент, що дозволяє відслідковувати цифрові сигнали у режимі реального часу. Його використання дало змогу перевірити правильність передачі даних на РК-дисплей, проконтролювати таймінги сигналів керування силовими елементами, а також відстежити імпульси з витратоміра. Це, у свою чергу, допомогло встановити, чи виходить проблема з боку мікроконтролера, чи з боку виконавчого модуля.

В рамках проєкту було детально вивчено апаратну структуру ПРК системи «Шельф», зокрема внутрішню логіку взаємодії між центральним контролером і периферійними модулями. У результаті цього аналізу вдалося отримати знання про специфічні схемотехнічні нюанси, властиві даній системі. Ці знання виявилися вкрай корисними під час пошуку несправностей у реальних умовах.

Крім цього, була отримана практична навичка збирання повної електронної системи ПРК — від монтажу окремих модулів до підключення роз'ємів і проведення пусконаладжувальних робіт. Такий досвід дозволив навчитися не тільки перевіряти якість монтажу, але й правильно зчитувати документацію, інтерпретувати електричні схеми, дотримуватись технічних вимог до з'єднань.

Таким чином, завдяки поєднанню практики, використання спеціалізованих діагностичних інструментів, глибокого аналізу

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 81   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

схемотехнічних нюансів та роботи з оригінальним обладнанням, було сформовано надійну базу практичних навичок для майбутньої роботи в галузі обслуговування, ремонту та модернізації паливороздавальних колонок (рис. 4.2).



Рисунок 4.2– Модулі які потрібно відремонтувати

#### 4.3. **Можливості модернізації та створення власних електронних модулів**

У ході розробки, налагодження та аналізу функціонування паливороздавального обладнання було виявлено ряд напрямів, у яких можлива подальша модернізація програмного та апаратного забезпечення. Ці вдосконалення не лише дозволять розширити функціональні можливості системи, а й значно підвищать зручність обслуговування та діагностики, а отже, ефективність у повсякденній роботі.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
| Змн. | Арк. | № документа | Підпис | Дата |                   | 82   |

Одним із ключових напрямів для модернізації є розширення програмного коду системи шляхом інтеграції з внутрішніми протоколами обміну даними, які використовує оригінальний виробник обладнання. Наразі програмне забезпечення реалізоване автономно й не пов'язане з протоколами зв'язку верхнього рівня (наприклад, для диспетчеризації, зчитування баз даних тощо). Однак у разі отримання офіційного дозволу на використання цих протоколів можливо реалізувати повну інтеграцію з центральною системою управління. Це дасть можливість дистанційного керування з боку системи управління, зокрема: автоматичного запуску/зупинки заправки, встановлення параметрів наливу, моніторингу поточного стану обладнання та його взаємодії з обліковими системами.

Крім цього, у поточній реалізації було помічено, що інтерфейс потребує вдосконалення з точки зору зручності та стабільності роботи. Зокрема, планується поліпшити тактильність управління — зробити інтерфейс більш чуйним до натискань, усунути затримки й оптимізувати навігацію по меню. Також є потреба у розширенні набору службових пунктів у меню, що будуть доступні через протокол обміну даними. Це дозволить інтегрувати нові функції без порушення базової логіки роботи системи.

Ще одним стратегічним напрямом є розробка власних електронних модулів, що будуть сумісні із системою. Зокрема, вже заплановано створення тестера для датчиків імпульсу — пристрою, який дозволить моделювати різні режими роботи витратоміра (рис. 4.3). В основі тестера передбачається використання крокового двигуна, що забезпечуватиме точне й регульоване генерування імпульсів, а також оптичного щілинного датчика, який імітуватиме реальну поведінку витратоміра. Такий підхід дозволить точно діагностувати проблеми — чи пов'язані вони з самими датчиками, чи з приймальним трактом ПРК. У результаті — значне скорочення часу на ремонт та збільшення точності діагностики.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 83   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

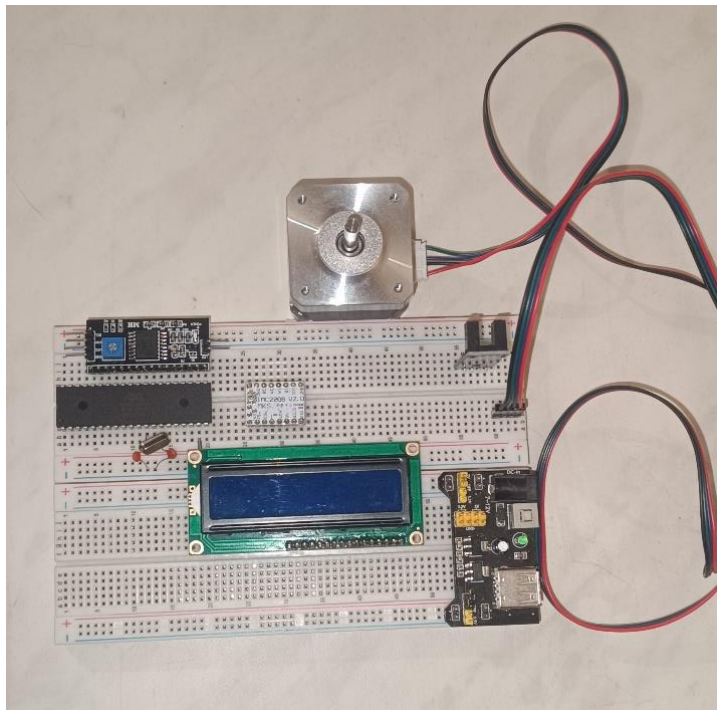


Рисунок 4.3– Модулі для тестеру датчика імпульсу

Окрім цього, для потреб підключення ПРК до комп'ютерних систем управління планується створення захищеного інтерфейсного перетворювача, що працюватиме на основі мікросхеми FT232RL. Особливістю цієї конструкції буде наявність гальванічної розв'язки, що забезпечить додатковий рівень захисту як для самого мікроконтролера, так і для зовнішніх пристроїв, зменшуючи ризик пошкоджень від перешкод або відмінностей потенціалів між пристроями. Цей пристрій дозволить здійснювати надійний обмін даними по інтерфейсу RS485–UART, з високою стійкістю до перешкод. Такий перетворювач дозволить безпечно підключати до системи керування.

Загалом, можливості для подальшої модернізації системи відкривають широке поле для розвитку як у напрямі розширення функціоналу, так і у сфері професійної діагностики та сервісного обслуговування. Реалізація цих ідей дозволить зробити обладнання більш зручним, стабільним і гнучким у використанні, а також підвищити ефективність роботи технічного персоналу.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 84   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

#### 4.4. Перспективи самостійної розробки апаратно-програмних рішень

Під час виконання дипломної роботи я здобув цінний досвід у розробці програмного забезпечення для мікроконтролера, роботі з апаратними модулями, оптимізації функціональних алгоритмів та налагодженні електронних систем. Важливим чинником стало також щоденне практичне застосування цих знань у професійній діяльності як сервісного інженера з обслуговування паливороздавальних колонок.

Завдяки глибокому аналізу схемотехнічних рішень, розумінню логіки функціонування ПРК, а також практичній взаємодії з усіма ключовими вузлами (силовими модулями, датчиками, інтерфейсами, системою індикації та керування) сформувалося цілісне уявлення про принципи побудови подібної техніки.

Усе це відкриває реальні перспективи для самостійної розробки повноцінного апаратно-програмного комплексу, що може стати альтернативою або покращеною версією існуючих систем ПРК. Наявність досвіду програмування низькорівневих драйверів, роботи з інтерфейсами (RS485, UART), реалізації меню, наливу, контролю помилок, а також практичні навички з діагностики та ремонту — дозволяють впевнено рухатись у напрямку розробки власної продукції.

У подальших етапах можливий перехід до створення власних друкованих плат, реалізації сумісного протоколу обміну даними, розробки більш надійної та зручної системи керування з урахуванням сучасних вимог. Таким чином, дипломна робота стала не лише навчальним проєктом, а й практичним фундаментом для розвитку повноцінного інженерного напрямку в галузі паливної електроніки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 85   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

## Висновки до розділу 4

У результаті виконання даної дипломної роботи було досягнуто вагомих практичних результатів, які не лише дозволили реалізувати функціональне програмне забезпечення для модуля керування паливороздавальною колонкою, а й заклали міцну основу для мого професійного розвитку як інженера-розробника.

Отримані навички програмування мікроконтролерів, зокрема використання асемблерних вставок у мові Сі, робота з апаратними модулями — аналоговим компаратором, таймерами, портами вводу/виводу — значно розширили мої технічні компетенції. Я навчився будувати модульну архітектуру коду, поєднувати програмні блоки між собою, забезпечуючи надійність та масштабованість системи.

Велике значення мала практична робота зі стендом, яка сприяла формуванню навичок діагностики та ремонту ПРК. Я навчився виявляти несправності у роботі датчиків, індикаторів, силових модулів та ефективно використовувати інструменти, зокрема логічний аналізатор, для аналізу цифрових сигналів та пошуку джерел помилок.

На основі отриманого досвіду були сформульовані конкретні напрямки можливої модернізації системи: інтеграція інтерфейсного перетворювача з гальванічною розв'язкою на базі FT232RL, створення стенду для тестування імпульсних датчиків, удосконалення меню та покращення стабільності керування. За наявності доступу до протоколу передачі даних з'явиться можливість реалізувати повноцінну інтеграцію з системою управління та реалізувати дистанційне керування.

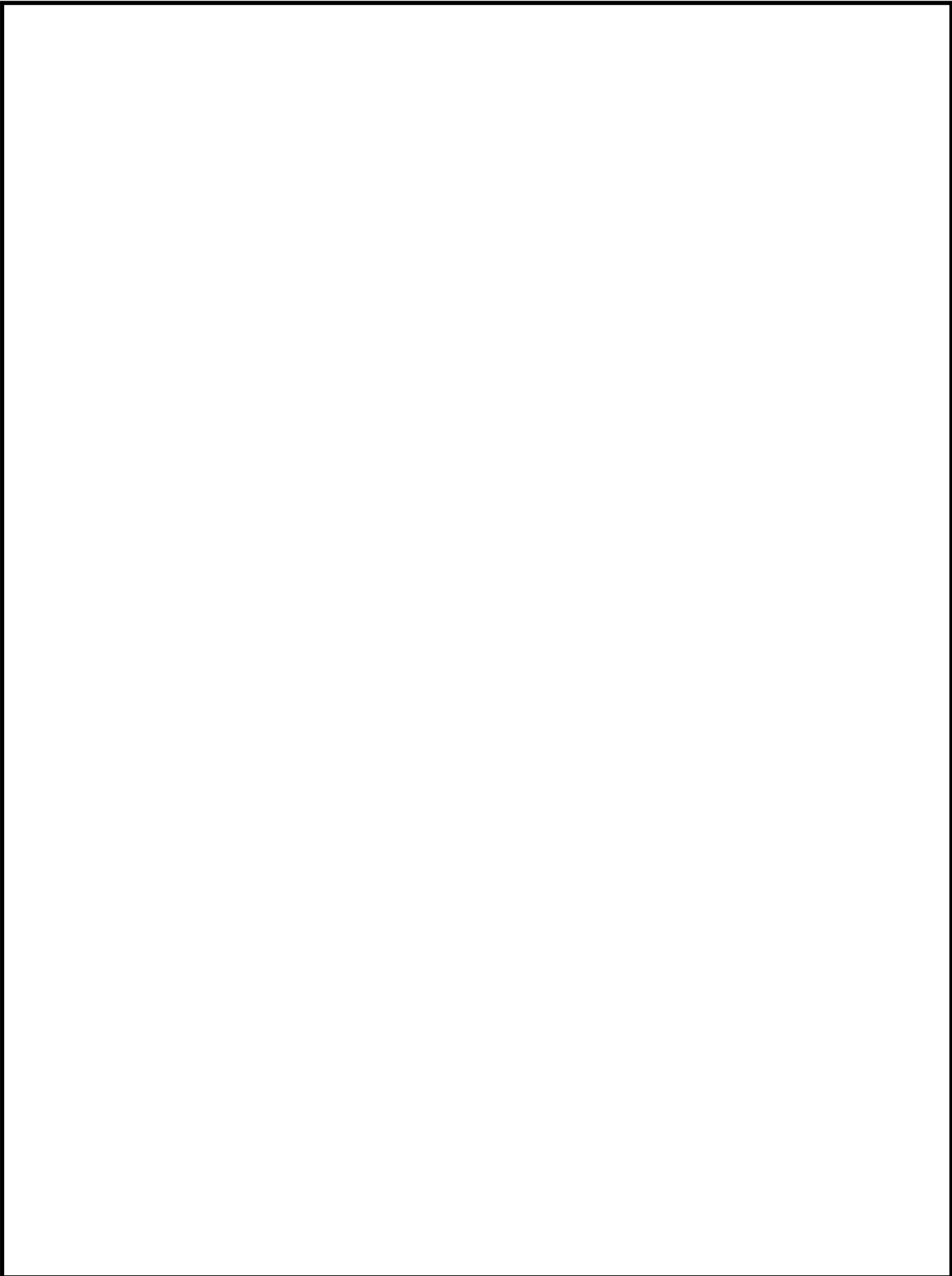
Отриманий досвід також відкриває перспективи повноцінної самостійної розробки апаратно-програмного рішення для ПРК, яке може стати надійною та сучасною альтернативою існуючим комерційним системам. Таким чином, дана робота має не лише навчальне, але й глибоке

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
| Змн. | Арк. | № документу | Підпис | Дата |                   | 86   |

прикладне значення, забезпечуючи базу для подальшого розвитку у сфері вбудованих систем та сервісного обслуговування паливної інфраструктури.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р4 | Арк. |
|      |      |             |        |      |                   | 87   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |





|             |      |                  |                                                                                     |      | 171.4321.КР.ПЗ.Р5 |                                |      |        |
|-------------|------|------------------|-------------------------------------------------------------------------------------|------|-------------------|--------------------------------|------|--------|
|             |      |                  |                                                                                     |      |                   |                                |      |        |
| Змн.        | Арк. | № документу      | Підпис                                                                              | Дата |                   |                                |      |        |
| Розробник   |      | Піряник А.Р.     |  |      | Охорона праці     | Літ.                           | Арк. | Аркуші |
| Консультант |      | Губальцев А. М.  |  |      |                   |                                | 88   |        |
| Н. контр.   |      | Добрінова Л. П.  |  |      |                   | НУК<br>імені адмірала Макарова |      |        |
| Керівник    |      | Жук О. К.        |  |      |                   |                                |      |        |
| Зав. каф.   |      | Рябенський В. М. |  |      |                   |                                |      |        |

## РОЗДІЛ 5. ОХОРОНА ПРАЦІ

### 5.1. Аналіз небезпечних факторів

Паливороздавальні колонки є складними технічними пристроями, що працюють в умовах підвищеного ризику. Їх експлуатація супроводжується впливом різноманітних небезпечних факторів — вибухових, хімічних, електричних, фізичних, ергономічних, які можуть нести загрозу як для персоналу, так і для споживачів. Аналіз цих ризиків є необхідним етапом при проектуванні, модернізації або обслуговуванні ПРК [15].

Одним з найнебезпечніших факторів є ризик виникнення пожежі або вибуху. Основними джерелами загрози тут виступають витoki палива під час заправки, іскроутворення через накопичення статичної електрики, використання відкритого вогню або несправне електрообладнання. Пари бензину або дизпалива, що накопичуються поблизу ПРК, можуть легко спалахнути навіть від невеликої іскри. Високу небезпеку становить відсутність або несправність вентиляції резервуарів, а також незаземлені металеві частини обладнання, через що виникає ризик електростатичного розряду. Рівень загрози в цьому випадку класифікується як високий, оскільки наслідки можуть бути катастрофічними — вибух, пожежа, ураження персоналу або клієнтів.

Реальні події підтверджують ці ризики. В червні 2015 року в Аккрі, столиці Гани[14]. Під час сильних дощів багато людей шукали притулку на автозаправній станції (рис. 5.1). Через витік пального та подальший вибух загинуло близько 150 осіб. Ця трагедія стала однією з наймасштабніших в історії країни .

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 89   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



Рисунок 5.1– Наслідки детонації та пожежі

З початку активних бойових дій зафіксовано десятки випадків цілеспрямованих атак на АЗС по всій території України, зокрема у прифронтових регіонах — Херсонській, Миколаївській (Очаків, Горохівка), Донецькій, Харківській областях (рис.5.2). Такі атаки супроводжуються масштабними пожежами, детонацією паливних резервуарів, руйнуванням будівель і, найгірше, — людськими жертвами серед працівників та клієнтів.



(а)



(б)



(в)

Рисунок 5.2– Наслідки атаки (а) АЗС в Херсоні, (б) АЗС в Очакові, нафтобази в Горохівці

Хімічні ризики пов'язані з постійним контактом персоналу із парами пального та іншими шкідливими речовинами. Вдихання парів бензину, що містять бензол, ксилол, толуол та інші токсичні компоненти, може призвести до хронічних захворювань дихальних шляхів, алергічних реакцій, дерматитів, а в довгостроковій перспективі — до онкологічних захворювань. Також небезпеку становить потрапляння пального на шкіру або в очі, особливо при обслуговуванні обладнання чи при нештатних ситуаціях.

Електричні ризики пов'язані із недоліками електроустановок або неправильним використанням обладнання. На АЗС часто трапляються випадки зношення ізоляції кабелів, використання невибухозахищених світильників у вибухонебезпечних зонах, несправність щитового обладнання. Відсутність заземлення або засобів автоматичного захисту від витoku струму може призвести до ураження електричним струмом, короткого замикання або іскроутворення, що в умовах наявності парів палива створює додаткову пожежонебезпеку.

Фізичні та механічні небезпеки найчастіше стосуються травмування персоналу внаслідок недбалості або конструктивних недоліків. Зокрема, це може бути наїзд автомобіля на працівника, падіння при обслуговуванні резервуарів, обрив паливного шланга, що залишився в баку. Частими є удари, падіння предметів, ковзання на вологій або обмерзлій поверхні, а також зіткнення транспорту з колонками. Незважаючи на те, що ці ризики мають локальний характер, саме вони найчастіше спричиняють виробничі травми.

Окрему групу становлять ергономічні та психосоціальні фактори. Персонал АЗС часто працює в умовах змінного графіку, включно з нічними змінами, що негативно впливає на біологічний ритм організму. Тривале стояння на ногах, робота в умовах холоду або спеки, контакт із великою кількістю клієнтів, шум, запахи — усе це створює додаткове навантаження на фізичний і психоемоційний стан працівника. Відсутність регулярного

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P5 | Арк. |
|      |      |             |        |      |                   | 91   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

навчання щодо дій у надзвичайних ситуаціях ще більше знижує рівень безпеки.

Основними рекомендаціями щодо зниження ризиків є: покращення вентиляційних систем, обов'язкове заземлення обладнання, використання вибухозахищених компонентів, встановлення протипожежних датчиків, регулярні інструктажі та навчання персоналу, а також впровадження організаційних заходів, що сприяють зниженню навантаження на працівників.

Воєнні загрози вимагають нових підходів до проектування, захисту та організації роботи ПРК, включаючи: створення додаткових захисних укриттів для персоналу та мінімізацію перебування персоналу в небезпечних зонах під час тривоги.

Комплексний підхід до аналізу та управління небезпеками дозволяє підвищити безпеку роботи ПРК, зменшити ризик аварійних ситуацій і створити безпечні умови праці для персоналу. Такий підхід особливо важливий у контексті модернізації застарілого обладнання, де заміна програмного забезпечення супроводжується необхідністю оцінки всіх супутніх ризиків.

## 5.2. Заходи щодо забезпечення електробезпеки

Електробезпека є одним із ключових аспектів загальної безпеки паливороздавальних колонок, особливо в умовах вибухонебезпечного середовища автозаправних станцій. Несправності в електрообладнанні, порушення правил експлуатації, або відсутність елементарного захисту можуть стати причиною пожежі, вибуху або ураження людей електричним струмом. У зв'язку з цим електричні системи мають бути спроектовані та експлуатовані з дотриманням найвищих стандартів безпеки (рис. 5.3).

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 92   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |



Рисунок 5.3– Автозарядна станція всередині

На автозаправних станціях функціонує складна електрична інфраструктура, яка живить широкий спектр обладнання. Основними споживачами є паливороздавальні колонки, що містять насоси, електромагнітні клапани, системи керування та дисплеї. Значну частину навантаження займають освітлення та вивіски, які необхідні для забезпечення безпеки на території станції у темний час доби. Також до складу електросистем входять електронні платіжні термінали, POS-системи, зчитувачі карток та системи обробки транзакцій. Окреме місце займають системи безпеки — камери відеоспостереження, сигналізація, контроль доступу.

Нестабільне та агресивне середовище АЗС підвищує ризик виникнення аварій внаслідок іскор, коротких замикань або витoku струму. Пари пального можуть загорітися навіть від слабкого джерела запалювання, тому кожен елемент системи повинен мати відповідний ступінь вибухозахисту. Усі

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P5 | Арк. |
|      |      |             |        |      |                   | 93   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |

електроустановки в зоні заправки мають бути спеціально сертифіковані для роботи у вибухонебезпечних умовах.

Важливим елементом електробезпеки є система заземлення та зрівнювання потенціалів. Усі металеві корпуси — від колонок до електрощитів — повинні бути надійно заземлені, щоб запобігти накопиченню електростатичних зарядів. Особливо це актуально під час перекачування пального, яке внаслідок тертя створює умови для виникнення електростатики. Крім того, необхідно регулярно перевіряти справність заземлюючих контурів та опору ізоляції.

До технічних заходів належать: встановлення реле захисного відключення (УЗО), автоматичних вимикачів, використання кабелів належного перерізу, герметизація зовнішніх з'єднань у захисній трубі, наявність ізоляційних килимків, діелектричних рукавиць та інструментів. Зони високої напруги повинні бути замкнені, позначені попереджувальними написами та недоступні для сторонніх осіб. Насоси рекомендується розміщувати на безпечній відстані від джерел живлення та основної електропроводки.

Слід регулярно проводити огляд та випробування всього електрообладнання. Виявлення зношених шнурів, несправних роз'ємів або пошкоджень має супроводжуватися негайним ремонтом або заміною відповідних елементів. Особливу увагу слід приділяти кабельним з'єднанням у місцях з підвищеною вологістю або ризиком механічного пошкодження.

До організаційних заходів належить систематичне навчання персоналу правилам електробезпеки та діям у разі аварійних ситуацій. Працівники повинні мати допуск до роботи в електроустановках відповідної категорії, проходити інструктажі та вміти знеструмлювати обладнання у разі потреби. Важливо також розміщувати попереджувальні знаки, наприклад, із заборороною користування мобільними телефонами під час заправки, оскільки навіть електромагнітне поле може бути джерелом загрози.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 94   |
| Змн. | Арк. | № документа | Підпис | Дата |                   |      |



Ефективним є впровадження додаткових систем захисту, таких як автоматичні датчики виявлення витоків пального, що у випадку загрози відразу відключають живлення. Окрему увагу слід приділяти розробці та впровадженню планів дій у надзвичайних ситуаціях — це дозволяє знизити паніку і мінімізувати наслідки аварії.

Таким чином, забезпечення електробезпеки на АЗС — це не лише встановлення технічних засобів захисту, а й впровадження системного підходу до планування, експлуатації та обслуговування електрообладнання. Поєднання регулярного технічного контролю, якісного навчання персоналу та відповідності вимогам нормативної документації дозволяє суттєво знизити ризики ураження струмом, пожежі або вибуху та створити безпечне середовище як для працівників, так і для відвідувачів автозаправних станцій.

### 5.3. Пожежна безпека при експлуатації обладнання

Пожежна безпека є одним із найважливіших напрямів безпечної експлуатації паливороздавальних колонок та обладнання автозаправних станцій (рис. 5.4). Основна небезпека полягає в постійній присутності легкозаймистих речовин та парів, які за певних умов можуть швидко спалахнути або спричинити вибух. Причинами займання можуть бути як технічні несправності, так і людський фактор — порушення правил користування, недбалість або відсутність належного контролю.



Рисунок 5.4— Засоби протипожежної боротьби [16]

Основні джерела пожежної небезпеки — це пари бензину, дизелю та газу, які можуть накопичуватись у резервуарах, технічних відсіках або в зоні роздачі пального. До потенційних причин займання належать: коротке замикання в електромережі, несправні насоси, відкриті електричні контакти, електростатичні розряди, відкритий вогонь, іскри від механічного обладнання або куріння в заборонених зонах. Особливо небезпечними є витоки палива, які можуть супроводжуватись розливом, випаровуванням і створенням вибухонебезпечного середовища.

Для зниження ризику пожежі всі елементи електрообладнання в зоні роздачі повинні бути вибухозахищеного виконання, а технічні з'єднання — герметичними. Необхідно передбачити регулярне технічне обслуговування обладнання, зокрема перевірку насосів, клапанів, кабельних трас, захисних автоматів і стану заземлення. Після кожного технічного обслуговування має вестися відповідний запис у журналі обліку.

Усі АЗС повинні бути обладнані первинними засобами пожежогасіння — порошковими, пінними та вуглекислотними вогнегасниками, які розміщуються на видимих та доступних місцях біля паливороздавальних колонок. Додатково встановлюється пожежний щит із набором ручних інструментів (лопати, багри, відра з піском). Вогнегасники перевіряються не рідше одного разу на рік. Ефективним рішенням є впровадження автоматичних систем пожежогасіння, які здатні виявляти загоряння та самостійно його ліквідовувати без участі персоналу, що суттєво скорочує час реагування.

Серед сучасних засобів пожежогасіння варто відзначити системи пінного гасіння, які швидко утворюють ізолювальний шар над палаючим паливом, а також системи водяного туману, що охолоджують зону загоряння, сповільнюючи поширення вогню та одночасно використовуючи мінімальну кількість води. Це особливо актуально в умовах, де надлишкова рідина може становити небезпеку (електричне обладнання, платформи управління).

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P5 | Арк. |
| Змн. | Арк. | № документу | Підпис | Дата |                   | 96   |

Важливим аспектом пожежної безпеки є організація просторової ізоляції — дотримання безпечних відстаней між паливороздавальними колонками та об'єктами, що можуть стати джерелами вогню. Згідно з сучасними рекомендаціями, АЗС повинні мати буферну зону від закладів підвищеної пожежної небезпеки, таких як автомайстерні, ресторани, об'єкти громадського харчування, склади горючих речовин тощо. Відомі випадки, коли вогонь із сусідньої будівлі швидко перекидався на заправку, що доводить критичну важливість просторової ізоляції.

Проектування АЗС також має бути підпорядковане принципам пожежної безпеки. Паливороздавальні колонки мають бути розташовані на безпечній відстані одна від одної, щоб у разі пожежі одна колонка не спричинила загоряння сусідньої. Слід використовувати негорючі або вогнестійкі матеріали в конструкціях дахів, стін, захисних кожухів. Доцільно передбачити вогнестійкі бар'єри, що відокремлюють критичні зони, наприклад, зони зберігання палива від адміністративних приміщень або кас.

Не менш важливою є організація дій персоналу в надзвичайних ситуаціях. Працівники повинні регулярно проходити протипожежний інструктаж, знати місця розміщення вогнегасників, автоматичних кнопок аварійного відключення, мати чіткий алгоритм дій у разі загоряння. Обов'язковим є наявність плану евакуації та пожежного щита, що мають бути розміщені в легкодоступних і видимих місцях.

Також варто звернути увагу на культурні та інформаційні аспекти. Слід заборонити використання мобільних телефонів у зоні ПРК, а також розмістити помітні попереджувальні знаки, що інформують про заборону куріння, користування відкритим вогнем і мобільними пристроями.

Комплексне впровадження технічних і організаційних заходів дозволяє суттєво знизити ризик виникнення пожежі або вибуху під час експлуатації обладнання. Пожежна безпека на АЗС — це не лише дотримання нормативів,

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 97   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

а й усвідомлена відповідальність персоналу, надійність технічної бази та передбачливість у проєктних рішеннях.

#### **5.4. Вимоги до організації робочого місця розробника**

Організація робочого місця розробника повинна відповідати вимогам охорони праці та забезпечувати безпечні умови під час виконання робіт з налагодження, діагностики та тестування електронних модулів паливороздавальних колонок. Особливо це стосується використання тестового стенду, у складі якого є силовий блок, що живиться від побутової електромережі 230 В, що підвищує ризик ураження електричним струмом.

Для зниження ризиків електротравм усі елементи стенду, які працюють під високою напругою, повинні бути ізольовані, що виключають можливість випадкового дотику до струмопровідних частин. Живлення стенду має здійснюватися через захищене джерело електроенергії із обов'язковим захисним заземленням і автоматичним вимикачем. Робоча поверхня повинна мати діелектричну підкладку.

При роботі з паяльним обладнанням необхідно дотримуватись правил пожежної безпеки. Робоче місце має бути оснащено витяжною або вентиляцією для видалення шкідливих парів. Поруч має знаходитися вогнегасник вуглекислотного типу, придатний для гасіння електрообладнання під напругою. Усі монтажні та паяльні роботи виконуються лише у знеструмленому стані, з попереднім перевірнням відсутності напруги. Також необхідно дотримуватися порядку зберігання електронних компонентів.

Враховуючи, що робота з електронікою вимагає високої зосередженості, робоче місце повинно бути добре освітлене, без зайвих джерел шуму та вібрацій. Висота стільця та стола повинна відповідати ергономічним нормам, щоб запобігти перенавантаженню опорно-рухового

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.KP.ПЗ.P5 | Арк. |
|      |      |             |        |      |                   | 98   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

апарату. При тривалій роботі за комп'ютером або з мікроскопом необхідно робити перерви для зниження навантаження на зір (рис. 5.5).



Рисунок 5.5– Організація зручного робочого місця [17]

Таким чином, дотримання вимог охорони праці під час організації робочого місця розробника є обов'язковою умовою безпечної та ефективної роботи з мікроконтролерними системами і силовими модулями у складі паливороздавальних комплексів.

### Висновки до розділу 5

Охорона праці є невід'ємною складовою безпечної експлуатації обладнання на автозаправних станціях, а також розробки, налагодження й тестування електронних систем, що керують роботою ПРК. У цьому розділі було розглянуто основні ризики, що виникають як при безпосередній роботі з паливними продуктами, так і при проектуванні та програмуванні мікроконтролерної техніки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 99   |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

Проведено аналіз небезпечних факторів, серед яких особливу увагу приділено пожежонебезпеці, вибуховості, токсичності пального, ризику ураження електричним струмом та ергономічним загрозам для персоналу. Окремо відзначено сучасні загрози, що виникають в умовах воєнного стану, зокрема прильоти боєприпасів по об'єктах АЗС, що актуалізує необхідність додаткових заходів фізичного захисту.

Для зниження ризиків визначено конкретні заходи електробезпеки: заземлення обладнання, використання засобів індивідуального захисту, встановлення ПЗВ (пристроїв захисного вимкнення), вибухозахищених елементів. Пожежна безпека розглянута з позицій організації вогнезахисту, наявності засобів виявлення та локалізації займання, а також навчання персоналу.

У підрозділі щодо організації робочого місця розробника визначено вимоги до безпечного, ергономічного і продуктивного середовища для програмування, налагодження та діагностики мікроконтролерних систем. Оскільки під час роботи використовуються живлення 230 В та високочутливі елементи, акцент зроблено на антистатичному захисті, вентиляції, якісному освітленні та використанні безпечного інструменту.

Загалом, реалізація усіх заходів, описаних у цьому розділі, дозволяє мінімізувати виробничі ризики та забезпечити безпечні умови праці для розробників і персоналу АЗС, що експлуатує паливороздавальні колонки.

|      |      |             |        |      |                   |      |
|------|------|-------------|--------|------|-------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Р5 | Арк. |
|      |      |             |        |      |                   | 100  |
| Змн. | Арк. | № документу | Підпис | Дата |                   |      |

## ВИСНОВКИ

У результаті виконання дипломної роботи за темою «Розробка мікроконтролерної системи наливання нафтопродуктів у складі паливороздавального комплексу» було реалізовано повноцінну програмну платформу для керування паливороздавальним модулем на основі мікроконтролера ATmega64, з урахуванням вимог до функціональності, безпеки та сумісності з апаратною частиною ПРК виробництва ТОВ «Шельф».

У ході роботи було проведено аналіз сучасних рішень, визначено вимоги до систем керування, обґрунтовано вибір апаратної платформи та сформульовано технічне завдання. На базі дослідження апаратної структури та особливостей схемотехнічного виконання плати керування, розроблено програмну архітектуру, що включає модулі обробки індикації, клавіатури, сенсорів, управління мотором і клапанами, а також реалізацію режимів заправки.

Особлива увага приділена тестуванню: створено діагностичний стенд, проведено повний цикл випробувань, виявлено та усунено критичні помилки, оптимізовано швидкодію системи. Було підтверджено відповідність реалізованого ПЗ вимогам до функціональності, стабільності та точності.

Окремо акцентовано на практичному значенні виконаної роботи: набуті навички в програмуванні, модульному проектуванні, роботі з асемблером, діагностиці, а також глибоке розуміння структури електроніки ПРК стали основою для подальшої професійної діяльності. Запропоновані шляхи модернізації, зокрема інтеграція інтерфейсного модуля з гальванічною розв'язкою та створення тестера імпульсних датчиків, підтверджують потенціал подальшого розвитку системи.

Завдяки дотриманню вимог охорони праці, аналізу ризиків та організації безпечного робочого середовища, проєкт виконано з урахуванням чинних нормативних вимог. Робота має не лише навчальне, а й прикладне

|      |      |             |        |      |                         |      |
|------|------|-------------|--------|------|-------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Висновки | Арк. |
|      |      |             |        |      |                         | 101  |
| Змн. | Арк. | № документу | Підпис | Дата |                         |      |



значення — її результати можуть бути використані для модернізації відомчих ПРК, продовження терміну служби застарілих пристроїв і створення альтернативних рішень в умовах обмеженого доступу до комерційного ПЗ.

|      |      |             |        |      |                         |      |
|------|------|-------------|--------|------|-------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Висновки | Арк. |
|      |      |             |        |      |                         | 102  |
| Змн. | Арк. | № документу | Підпис | Дата |                         |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tokheim Quality Calculator – Installation Manual. Scribd : вебсайт. URL: <https://scribd.com/document/516847520/Tokheim-Quality-Calculator-TQC-Installation-Maintenance-Manual> (дата звернення: 11.04.2025).
2. Принципова схема підключення. Shelf : вебсайт. URL: <https://shelf.ua/docs/manual/ps200.pdf> (дата звернення: 11.04.2025).
3. Контролери ПРК на АЗС. Нако : вебсайт. URL: <https://hako.com.ua/products/kontroleri-prk-na-azs/> (дата звернення: 15.04.2025).
4. Чабанний В. Я. Паливо-мастильні матеріали, технічні рідини та системи їх забезпечення : підручник / В. Я. Чабанний, С. О. Магопець, І. М. Осипов, В. В. Аулін, О. Й. Мажейка, Є. К. Солових, В. А. Павлюк-Мороз, Г. А. Попов, С. Є. Катеринич. – Кіровоград : Центрально-Українське видавництво, 2008. – 500 с. (дата звернення: 15.04.2025)
5. Fuel Dispenser Guide Book ZC. Scribd : вебсайт. URL: <https://ru.scribd.com/document/712037499/Fuel-Dispenser-Guide-Book-ZC-C> (дата звернення: 03.05.2025).
6. Комплектуючі ПРК. Shelf : вебсайт. URL: <https://shelf.ua/products/zip/dispenser-zip/> (дата звернення: 11.04.2025).
7. Електронні комплектуючі ПРК. Shelf : вебсайт. URL: <https://shelf.ua/products/zip/electronics-zip/> (дата звернення: 11.04.2025).
8. Fuel Dispenser: Manual Instruction. Scribd : вебсайт. URL: <https://ru.scribd.com/document/589648729/Instruction> (дата звернення: 10.05.2025).
9. Fuel Dispensing Technology. SlideShare : вебсайт. URL: <https://www.slideshare.net/slideshow/fuel-dispensing-technology/75898501> (дата звернення: 11.05.2025).
10. Паливороздавльні колонки серії «Шельф»100 та «Шельф» 100ВК. Shelf : вебсайт. URL: <https://shelf.ua/products/eqazs/s100/> (дата звернення: 11.04.2025).

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Джерела | Арк. |
|      |      |             |        |      |                        | 103  |
| Змн. | Арк. | № документу | Підпис | Дата |                        |      |

11. ATMEGA64 Datasheet .alldatasheet : вебсайт. URL: <https://www.alldatasheet.com/datasheet-pdf/view/255851/ATMEL/ATMEGA64.html> (дата звернення: 11.04.2025).

12. AVR програматор USBasp. Mini-Tech : вебсайт. URL: <https://www.mini-tech.com.ua/ua/avr-programmator-usbasp> (дата звернення: 25.05.2025).

13. Логічний аналізатор 24МГц 8 каналів. Uamper : вебсайт. URL: <https://uamper.com/USB-Logic-Analyzer-24Mhz-8-CH> (дата звернення: 15.05.2025).

14. Пожежа на АЗС у Гані: загинуло понад 150 людей. BBC : вебсайт. URL: [https://www.bbc.com/ukrainian/ukraine\\_in\\_russian/2015/06/150605\\_ru\\_s\\_ghana\\_petrol\\_station\\_kills\\_about\\_175](https://www.bbc.com/ukrainian/ukraine_in_russian/2015/06/150605_ru_s_ghana_petrol_station_kills_about_175) (дата звернення: 25.05.2025).

15. Development of Fuel Dispenser Electronics. DergiPark : вебсайт. URL: <https://dergipark.org.tr/en/download/article-file/3822673> (дата звернення: 25.05.2025).

16. Пожежний щит – стенд. Euroservis : вебсайт. URL: <https://euroservis.com.ua/ua/pozharnyy-shchit-stend/> (дата звернення: 25.05.2025).

17. Робочий стіл. Ratelist : вебсайт. URL: <https://files.ratelist.top/uploads/images/bs/67511/photos/b9da2bd59ef3cb13db4de20296b0138e-original.webp> (дата звернення: 25.05.2025).

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Джерела | Арк. |
|      |      |             |        |      |                        | 104  |
| Змн. | Арк. | № документу | Підпис | Дата |                        |      |

## ДОДАТОК А

### Центральний код main.c

```
#define F_CPU 7372800UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define Dallas PA4
#define BUZ PA5
#define DATA_PIN PA6
#define SCK_PIN PA7
#define KOL1 PF0
#define KOL2 PF1
#define KOL3 PF2
#define KOL4 PF3
#define STR1 PA0
#define STR2 PA1
#define STR3 PA2
#define STR4 PA3
#define MOT PD7
#define VCUT PC1
#define VLOW PG0
#define TIME_50MS 2
#define TIME_100MS 3
#define TIME_150MS 5
void refueling() ;
void set_buzzer_mode(uint8_t mode);
uint8_t top_line_flag=11;
volatile uint8_t buz_flag = 0;
volatile uint8_t buz_state = 0;
volatile uint16_t buz_counter = 0;
volatile uint8_t pulse_count = 0;
volatile uint16_t pwm_counter = 0;
volatile uint8_t but_c_flag=0;
uint8_t field_flag=2;
uint32_t cost = 2222;
uint32_t cost_order=0;
uint32_t volume_now = 0;
uint32_t volume_old=500;
uint32_t volume_order=0;
volatile uint16_t edgeCount = 0;
volatile uint16_t capturedValue=0;
uint8_t dallas_flag=0;
uint8_t pass_flag7=0;
uint8_t pass_flag5=0;
uint8_t pass_flag3=0;
uint8_t menu_num=1;
uint8_t menu_muv=1;
uint8_t nozzle_flag=3;
uint32_t price = 8888888;
uint8_t shift = 0;
uint32_t volume_ol =1234567;
uint32_t cost_ol = 9999999;
uint8_t protocol7 =12;
uint8_t address =10;
uint8_t num_taps9=1;
uint8_t oper_mode10=17;
uint8_t speed_uart11=19;
uint8_t keys12=1;
uint8_t valve_act_delay13=33;
```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 105  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

uint8_t timeout_bet_launches14=22;
uint8_t timeout_bet_pulses15=1;
uint8_t retard_valve_adv16=10;
uint8_t impulse_magnit17=10;
uint8_t after_point18=1;
uint8_t rounding_mode19=22;
uint8_t hose_inflation20=1;
uint32_t calibr21=1990;
uint32_t calibration_cap=1990;
uint16_t timer1=0;
uint8_t SLOWDOWN_THRESHOLD=100;
volatile uint8_t pressed_button=17;
uint8_t number[22] = {};
uint8_t menu1[]={};
uint8_t menu2[]={};
uint8_t menu3[]={ };
uint8_t menu4[]={};
uint8_t menu5[]={};
uint8_t menu6[]={};
uint8_t menu7[]={};
uint8_t menu8[]={};
uint8_t menu9[]={};
uint8_t menu10[]={};
uint8_t menu11[]={};
uint8_t menu12[]={};
uint8_t menu13[]={};
uint8_t menu14[]={};
uint8_t menu15[]={};
uint8_t menu16[]={};
uint8_t menu17[]={};
uint8_t menu18[]={};
uint8_t menu19[]={};
uint8_t menu20[]={};
uint8_t menu21[]={};
uint8_t menu22[]={};
extern void process_menu0(uint8_t *num1);
extern void process_pass(void);
extern void start_sck1(void);
extern void start_sck2(void);
extern void send_packet(void);
extern void send_digit22(uint8_t digit);
extern void send_lit(uint8_t digit);
extern void Vers(void);
extern void process_digits1(uint8_t *digits);
void menu_inil(){
    if (menu_num==1&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (num_taps9<9999999)
            {
                price+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (num_taps9>1)
            {
                price-=1;
                menu_muv=1;
            }
        }
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 106  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

    }
    }
    uint32_t price_now=price;
    for (uint8_t i = 0; i < 16; i++){
        menu1[i]=10;
    }
    menu1[0]=1;
    menu1[1]=10;
    menu1[2]=top_line_flag;
    if (price_now < 100) {
        menu1[3] = price_now % 10;
        if (price_now >= 10) {
            price_now /= 10;
            menu1[4] = price_now % 10;
        } else {
            menu1[4] = 0;
        }
        menu1[5] = 0;
        for (uint8_t i = 7; i < 11; i++) {
            menu1[i] = 10;
        }
    }
    else {
        for (uint8_t i = 3; i < 11; i++) {
            if (price_now > 0) {
                menu1[i] = price_now % 10;
                price_now /= 10;
            } else {
                menu1[i] = 10;
            }
        }
    }
    menu1[11]=1;
    menu1[12]=19;
}
void menu_ini2(){
    for (uint8_t i = 0; i < 16; i++){
        menu2[i]=10;
    }
    menu2[0]=2;
    menu2[1]=10;
    menu2[2]=top_line_flag;
    menu2[3]=11;
    menu2[4]=10;
    menu2[5]=14;
}

void menu_ini3(){
    uint32_t volume_ol_now=volume_ol;
    for (uint8_t i = 0; i < 16; i++){
        menu3[i]=10;
    }
    if (nozzle_flag)
    {
        menu3[0]=1;
        menu3[1]=0;
        menu3[2]=14;
    }else{
        menu3[0]=3;
        menu3[1]=10;
        menu3[2]=top_line_flag;
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 107  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

    }
    if (volume_ol_now < 100) {
        menu3[3] = volume_ol_now % 10;
        if (volume_ol_now >= 10) {
            volume_ol_now /= 10;
            menu3[4] = volume_ol_now % 10;
        } else {
            menu3[4] = 0;
        }
        menu3[5] = 0;
        for (uint8_t i = 6; i < 17; i++) {
            menu3[i] = 10;
        }
    }else {
        for (uint8_t i = 3; i < 17; i++){
            if (volume_ol_now > 0) {
                menu3[i] = volume_ol_now % 10;
                volume_ol_now /= 10;
            } else {
                menu3[i] = 10;
            }
        }
    }
}

void menu_ini4(){
    uint32_t volume_ol_now=volume_ol;
    for (uint8_t i = 0; i < 16; i++){
        menu4[i]=10;
    }
    if (nozzle_flag)
    {
        menu4[0]=1;
        menu4[1]=0;
        menu4[2]=14;
    }else{
        menu4[0]=4;
        menu4[1]=10;
        menu4[2]=top_line_flag;
    }
    if (volume_ol_now < 100) {
        menu4[3] = volume_ol_now % 10;
        if (volume_ol_now >= 10) {
            volume_ol_now /= 10;
            menu4[4] = volume_ol_now % 10;
        } else {
            menu4[4] = 0;
        }
        menu4[5] = 0;
        for (uint8_t i = 6; i < 17; i++) {
            menu4[i] = 10;
        }
    }else {
        for (uint8_t i = 3; i < 17; i++){
            if (volume_ol_now > 0) {
                menu4[i] = volume_ol_now % 10;
                volume_ol_now /= 10;
            } else {
                menu4[i] = 10;
            }
        }
    }
}
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 108  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |



```

void menu_ini5(){
    uint32_t cost_ol_now=cost_ol;
    for (uint8_t i = 0; i < 16; i++){
        menu5[i]=10;
    }
    if (nozzle_flag)
    {
        menu5[0]=1;
        menu5[1]=0;
        menu5[2]=14;
    }else{
        menu5[0]=5;
        menu5[1]=10;
        menu5[2]=top_line_flag;
    }
    if (cost_ol_now < 100) {
        menu5[3] = cost_ol_now % 10;
        if (cost_ol_now >= 10) {
            cost_ol_now /= 10;
            menu5[4] = cost_ol_now % 10;
        } else {
            menu5[4] = 0;
        }
        menu5[5] = 0;
        for (uint8_t i = 6; i < 17; i++) {
            menu5[i] = 10;
        }
    }else {
        for (uint8_t i = 3; i < 17; i++){
            if (cost_ol_now > 0) {
                menu5[i] = cost_ol_now % 10;
                cost_ol_now /= 10;
            } else {
                menu5[i] = 10;
            }
        }
    }
}

void menu_ini6(){
    uint32_t volume_ol_now=777777;
    for (uint8_t i = 0; i < 16; i++){
        menu6[i]=10;
    }
    if (nozzle_flag)
    {
        menu6[0]=1;
        menu6[1]=0;
        menu6[2]=14;
    }else{
        menu6[0]=6;
        menu6[1]=10;
        menu6[2]=top_line_flag;
    }
    if (volume_ol_now < 100) {
        menu6[3] = volume_ol_now % 10;
        if (volume_ol_now >= 10) {
            volume_ol_now /= 10;
            menu6[4] = volume_ol_now % 10;
        } else {
            menu6[4] = 0;
        }
    }
    menu6[5] = 0;
    for (uint8_t i = 6; i < 17; i++) {

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 109  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

        menu6[i] = 10;
    }
    }else {
    for (uint8_t i = 3; i < 17; i++){
        if (volume_ol_now > 0) {
            menu6[i] = volume_ol_now % 10;
            volume_ol_now /= 10;
        } else {
            menu6[i] = 10;
        }
    }
    }
}

void menu_ini7(){
    if (menu_num==7&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (protocol7<=13)
            {
                protocol7+=1;
            }
            menu_muv=1;
        }
        if (menu_muv==0)
        {
            if (protocol7>=13)
            {
                protocol7-=1;
            }
            menu_muv=1;
        }
    }
    for (uint8_t i = 0; i < 16; i++){
        menu7[i]=10;
    }
    menu7[0]=7;
    menu7[1]=10;
    menu7[2]=top_line_flag;

    menu7[3]=protocol7;
    menu7[4]=10;
    menu7[5]=18;

}

void menu_ini8(){
    if (menu_num==8&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (addres8<250)
            {
                addres8+=10;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (addres8>10)
            {
                addres8-=10;
                menu_muv=1;
            }
        }
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 110  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

    }
    }
    uint8_t address8_now=address8;
    for (uint8_t i = 0; i < 16; i++){
        menu8[i]=10;
    }
    menu8[0]=8;
    menu8[1]=10;
    menu8[2]=top_line_flag;

    menu8[3]=10;
    menu8[4]=10;
    menu8[5]=10;
    for (uint8_t i = 6; i < 11; i++){
        if (address8_now > 0) {
            menu8[i] = address8_now % 10;
            address8_now /= 10;
        } else {
            menu8[i] = 10;
        }
    }

    menu8[11]=10;
    menu8[12]=18;

}

void menu_ini9(){
    if (menu_num==9&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (num_taps9<5)
            {
                num_taps9+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (num_taps9>1)
            {
                num_taps9-=1;
                menu_muv=1;
            }
        }
    }

    for (uint8_t i = 0; i < 16; i++){
        menu9[i]=10;
    }
    menu9[0]=9;
    menu9[1]=10;
    menu9[2]=top_line_flag;
    menu9[3]=10;
    menu9[4]=10;
    menu9[5]=10;
    menu9[6]=num_taps9;
    menu9[7]=10;
    menu9[8]=10;
    menu9[9]=10;
    menu9[10]=10;

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 111  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

        menu9[11]=10;
        menu9[12]=18;

    }
    void menu_ini10(){
        if (menu_num==10&&menu_muv!=1)
        {
            if (menu_muv==2)
            {
                if (oper_model0<=16)
                {
                    oper_model0+=1;
                }

                menu_muv=1;
            }
            if (menu_muv==0)
            {
                if (oper_model0>=16)
                {
                    oper_model0-=1;
                }
                menu_muv=1;
            }
        }
        for (uint8_t i = 0; i < 16; i++){
            menu10[i]=10;
        }
        menu10[0]=0;
        menu10[1]=1;
        menu10[2]=top_line_flag;
        menu10[3]=oper_model0;
        menu10[4]=10;
        menu10[5]=18;

    }
    void menu_ini11(){
        if (menu_num==11&&menu_muv!=1)
        {
            if (menu_muv==2)
            {
                if (speed_uart11<=19)
                {
                    speed_uart11+=1;
                }

                menu_muv=1;
            }
            if (menu_muv==0)
            {
                if (speed_uart11>=19)
                {
                    speed_uart11-=1;
                }
                menu_muv=1;
            }
        }
        for (uint8_t i = 0; i < 16; i++){
            menu11[i]=10;
        }
        menu11[0]=1;
        menu11[1]=1;
        menu11[2]=top_line_flag;

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 112  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

    menu11[3]=speed_uart11;
    menu11[4]=10;
    menu11[5]=18;

}

void menu_ini12(){
    for (uint8_t i = 0; i < 16; i++){
        menu12[i]=10;
    }
    menu12[0]=2;
    menu12[1]=1;
    menu12[2]=top_line_flag;
    menu12[3]=21;
    menu12[4]=1;
    menu12[5]=15;

}

void menu_ini13(){
    if (menu_num==13&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (valve_act_delay13<99)
            {
                valve_act_delay13+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (valve_act_delay13>0)
            {
                valve_act_delay13-=1;
                menu_muv=1;
            }
        }
    }

    uint8_t valve_act_delay13_now=valve_act_delay13;
    for (uint8_t i = 0; i < 16; i++){
        menu13[i]=10;
    }
    menu13[0]=3;
    menu13[1]=1;
    menu13[2]=top_line_flag;

    menu13[3]=10;
    menu13[4]=10;
    menu13[5]=10;
    for (uint8_t i = 6; i < 11; i++){
        if (valve_act_delay13_now > 0) {
            menu13[i] = valve_act_delay13_now % 10;
            valve_act_delay13_now /= 10;
        } else {
            menu13[i] = 10;
        }
    }

    menu13[11]=10;
    menu13[12]=18;
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 113  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

void menu_ini14(){
    if (menu_num==14&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (timeout_bet_launches14<180)
            {
                timeout_bet_launches14+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (timeout_bet_launches14>1)
            {
                timeout_bet_launches14-=1;
                menu_muv=1;
            }
        }
    }
    uint8_t timeout_bet_launches14_now=timeout_bet_launches14;
    for (uint8_t i = 0; i < 16; i++){
        menu14[i]=10;
    }
    menu14[0]=4;
    menu14[1]=1;
    menu14[2]=top_line_flag;

    menu14[3]=10;
    menu14[4]=10;
    menu14[5]=10;
    for (uint8_t i = 6; i < 11; i++){
        if (timeout_bet_launches14_now > 0) {
            menu14[i] = timeout_bet_launches14_now % 10;
            timeout_bet_launches14_now /= 10;
        } else {
            menu14[i] = 10;
        }
    }

    menu14[11]=10;
    menu14[12]=18;

}

void menu_ini15(){
    if (menu_num==15&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (timeout_bet_pulses15<60)
            {
                timeout_bet_pulses15+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (timeout_bet_pulses15>1)
            {
                timeout_bet_pulses15-=1;
                menu_muv=1;
            }
        }
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 114  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

    }

    }

    }
uint8_t timeout_bet_pulses15_now=timeout_bet_pulses15;
for (uint8_t i = 0; i < 16; i++){
    menu15[i]=10;
}
menu15[0]=5;
menu15[1]=1;
menu15[2]=top_line_flag;

menu15[3]=10;
menu15[4]=10;
menu15[5]=10;
for (uint8_t i = 6; i < 11; i++){
    if (timeout_bet_pulses15_now > 0) {
        menu15[i] = timeout_bet_pulses15_now % 10;
        timeout_bet_pulses15_now /= 10;
    } else {
        menu15[i] = 10;
    }
}

menu15[11]=10;
menu15[12]=18;

}

void menu_inil6(){
    if (menu_num==16&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (retard_valve_adv16<2000)
            {
                retard_valve_adv16+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (retard_valve_adv16>10)
            {
                retard_valve_adv16-=1;
                menu_muv=1;
            }
        }
    }
}

uint8_t retard_valve_adv16_now=retard_valve_adv16;
for (uint8_t i = 0; i < 16; i++){
    menu16[i]=10;
}
menu16[0]=6;
menu16[1]=1;
menu16[2]=top_line_flag;

if (retard_valve_adv16_now < 100) {
    menu16[3] = retard_valve_adv16_now % 10;
    if (retard_valve_adv16_now >= 10) {
        retard_valve_adv16_now /= 10;
        menu16[4] = retard_valve_adv16_now % 10;
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 115  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

        } else {
            menu16[4] = 0;
        }
        menu16[5] = 0;
        for (uint8_t i = 6; i < 17; i++) {
            menu16[i] = 10;
        }
    } else {
        for (uint8_t i = 3; i < 17; i++){
            if (retard_valve_adv16_now > 0) {
                menu16[i] = retard_valve_adv16_now % 10;
                retard_valve_adv16_now /= 10;
            } else {
                menu16[i] = 10;
            }
        }
    }
}

void menu_ini17(){
    if (menu_num==17&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (impulse_magnit17<1000)
            {
                impulse_magnit17+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (impulse_magnit17>1)
            {
                impulse_magnit17-=1;
                menu_muv=1;
            }
        }
    }
    uint8_t impulse_magnit17_now=impulse_magnit17;
    for (uint8_t i = 0; i < 16; i++){
        menu17[i]=10;
    }
    menu17[0]=7;
    menu17[1]=1;
    menu17[2]=top_line_flag;

    menu17[3]=10;
    menu17[4]=10;
    menu17[5]=10;
    for (uint8_t i = 6; i < 11; i++){
        if (impulse_magnit17_now > 0) {
            menu17[i] = impulse_magnit17_now % 10;
            impulse_magnit17_now /= 10;
        } else {
            menu17[i] = 10;
        }
    }

    menu17[11]=10;
    menu17[12]=18;

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 116  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |



```

}
void menu_ini18(){
    if (menu_num==18&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (after_point18<4)
            {
                after_point18+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (after_point18>0)
            {
                after_point18-=1;
                menu_muv=1;
            }
        }
    }
    for (uint8_t i = 0; i < 16; i++){
        menu18[i]=10;
    }
    menu18[0]=8;
    menu18[1]=1;
    menu18[2]=top_line_flag;

    menu18[3]=10;
    menu18[4]=10;
    menu18[5]=10;
    menu18[6]=after_point18;
    menu18[7]=10;
    menu18[8]=10;
    menu18[9]=10;
    menu18[10]=10;
    menu18[11]=10;
    menu18[12]=18;

}
void menu_ini19(){
    if (menu_num==19&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            rounding_mode19=23;
            menu_muv=1;
        }
        if (menu_muv==0)
        {
            rounding_mode19=22;
            menu_muv=1;
        }
    }
    for (uint8_t i = 0; i < 16; i++){
        menu19[i]=10;
    }
    menu19[0]=9;
    menu19[1]=1;
    menu19[2]=top_line_flag;
    menu19[3]=rounding_mode19;

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 117  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

menu19[4]=10;
menu19[5]=18;

}
void menu_ini20(){
    if (menu_num==20&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (hose_inflation20<100)
            {
                hose_inflation20+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (hose_inflation20>0)
            {
                hose_inflation20-=1;
                menu_muv=1;
            }
        }
    }
    uint8_t hose_inflation20_now=hose_inflation20;
    for (uint8_t i = 0; i < 16; i++){
        menu20[i]=10;
    }
    menu20[0]=0;
    menu20[1]=2;
    menu20[2]=top_line_flag;

    if (hose_inflation20_now < 100) {
        menu20[3] = hose_inflation20_now % 10;
        if (hose_inflation20_now >= 10) {
            hose_inflation20_now /= 10;
            menu20[4] = hose_inflation20_now % 10;
        } else {
            menu20[4] = 0;
        }
        menu20[5] = 0;
        for (uint8_t i = 6; i < 17; i++) {
            menu20[i] = 10;
        }

        menu20[11]=10;
        menu20[12]=18;
    }
}
void menu_ini21(){
    uint32_t calibr21_now=calibr21;
    for (uint8_t i = 0; i < 16; i++){
        menu21[i]=10;
    }

    menu21[0]=1;
    menu21[1]=2;
    menu21[2]=top_line_flag;

    if (calibr21_now < 100) {

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 118  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

menu21[3] = calibr21_now % 10;
if (calibr21_now >= 10) {
    calibr21_now /= 10;
    menu21[4] = calibr21_now % 10;
} else {
    menu21[4] = 0;
}
menu21[5] = 0;
for (uint8_t i = 6; i < 11; i++) {
    menu21[i] = 10;
}
}else {
for (uint8_t i = 3; i < 11; i++){
    if (calibr21_now > 0) {
        menu21[i] = calibr21_now % 10;
        calibr21_now /= 10;
    } else {
        menu21[i] = 10;
    }
}
}
menu21[11]=1;
menu21[12]=12;

}

void menu_ini22(){
    if (menu_num==22&&menu_muv!=1)
    {
        if (menu_muv==2)
        {
            if (calibration_cap<2000)
            {
                calibration_cap+=1;
                menu_muv=1;
            }
        }
        if (menu_muv==0)
        {
            if (calibration_cap>0)
            {
                calibration_cap-=1;
                menu_muv=1;
            }
        }
    }
}
uint32_t calibration_cap_now=calibration_cap;
for (uint8_t i = 0; i < 16; i++){
    menu22[i]=10;
}

menu22[0]=2;
menu22[1]=2;
menu22[2]=top_line_flag;

if (calibration_cap_now < 100) {
    menu22[3] = calibration_cap_now % 10;
    if (calibration_cap_now >= 10) {
        calibration_cap_now /= 10;
        menu22[4] = calibration_cap_now % 10;
    } else {
        menu22[4] = 0;
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 119  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

        menu22[5] = 0;
        for (uint8_t i = 6; i < 11; i++) {
            menu22[i] = 10;
        }
    }else {
        for (uint8_t i = 3; i < 11; i++){
            if (calibration_cap_now > 0) {
                menu22[i] = calibration_cap_now % 10;
                calibration_cap_now /= 10;
            } else {
                menu22[i] = 10;
            }
        }
    }
    menu22[11]=1;
    menu22[12]=12;
}

void AC_ini(){
    DDRE &= ~(1 << PE2) & ~(1 << PE3);
    ACSR = (1 << ACIE) | (0 << ACIS1) | (0 << ACIS0);
    sei();
}

void litres_ini() {
    DDRD &= ~(1 << PD6);
    PORTD &= ~(1 << PD6);

    TCCR1A = 0;
    TCCR1B = 0;
    TIMSK &= ~(1 << OCIE1A);
    TCCR1B = (1 << CS12) | (1 << CS11) | (0 << CS10);
}

void clear_array(uint8_t* array, uint8_t size) {
    for(uint8_t i = 1; i < size; i++) {
        array[i] = 10;
    }
}

void parse_digits(uint32_t price, uint8_t *menu0) {
    for (uint8_t i = 1; i < 4; i++) {
        if (price > 0) {
            menu0[i] = price % 10;
            price /= 10;
        } else {
            menu0[i] = 10;
        }
    }
}

void parse_digits_vp3(uint32_t volume, uint8_t *menu0) {
    for (uint8_t i = 1; i < 15; i++) {
        if (volume > 0) {
            menu0[i] = volume % 10;
            volume /= 10;
        } else {
            menu0[i] = 10;
        }
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 120  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

void parse_number_to_digits(uint32_t cost, uint32_t volume, uint16_t
price, uint8_t *number) {
    uint8_t index = 0;
    if (cost < 100) {
        number[index++] = cost % 10;
        if (cost >= 10) {
            cost /= 10;
            number[index++] = cost % 10;
        } else {
            number[index++] = 0;
        }
        number[index++] = 0;
        for (uint8_t i = 0; i < 5; i++) {
            number[index++] = 10;
        }
    }
    else {
        for (uint8_t i = 0; i < 8; i++) {
            if (cost > 0) {
                number[index++] = cost % 10;
                cost /= 10;
            } else {
                number[index++] = 10;
            }
        }
    }
    if (volume < 100) {
        number[index++] = volume % 10;
        if (volume >= 10) {
            volume /= 10;
            number[index++] = volume % 10;
        } else {
            number[index++] = 0;
        }
        number[index++] = 0;
        for (uint8_t i = 0; i < 5; i++) {
            number[index++] = 10;
        }
    }
    else {
        for (uint8_t i = 0; i < 8; i++) {
            if (volume > 0) {
                number[index++] = volume % 10;
                volume /= 10;
            } else {
                number[index++] = 10;
            }
        }
    }
    if (price < 100) {
        number[index++] = price % 10;
        if (price >= 10) {
            price /= 10;
            number[index++] = price % 10;
        } else {
            number[index++] = 0;
        }
        number[index++] = 0;
        for (uint8_t i = 0; i < 3; i++) {
            number[index++] = 10;
        }
    }
    else {

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 121  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

    for (uint8_t i = 0; i < 6; i++) {
        if (price > 0) {
            number[index++] = price % 10;
            price /= 10;
        } else {
            number[index++] = 10;
        }
    }
}

}

void start_paket() {
    PORTA |= (1 << DATA_PIN);
    _delay_ms(500);
    PORTA &= ~(1 << DATA_PIN);
    _delay_ms(310);
    PORTA |= (1 << DATA_PIN);
    start_sck1();
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    _delay_ms(600);

    PORTA &= ~(1 << DATA_PIN);
    start_sck2();
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    _delay_ms(600);

    for (int i = 9; i >= 0; i--) {
        send_digit22(i);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);
        _delay_ms(600);
    }
    start_sck2();
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    _delay_ms(95);
    Vers();
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    _delay_ms(20);

    for (int i = 12; i >= 0; i--) {
        Vers();
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);
        _delay_ms(250);
    }
}

ISR(TIMER1_COMPA_vect) {
    pwm_counter++;

    if (pwm_counter == 1) {
        PORTF |= (1 << KOL2) | (1 << KOL3) | (1 << KOL4);
    }
    if (pwm_counter == 75) {
        PORTF ^= (1 << KOL4) | (1 << KOL1);
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 122  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

if (pwm_counter == 76) {
    PORTF ^= ((1 << KOL4) | (1 << KOL3));
}
if (pwm_counter == 77) {
    PORTF ^= ((1 << KOL3) | (1 << KOL2));
}
if (pwm_counter == 78) {
    PORTF ^= ((1 << KOL2) | (1 << KOL1));
    pwm_counter = 0;
}

if(PINA & ((1 << STR1) | (1 << STR2) | (1 << STR3) | (1 << STR4))) {
    but_c_flag=1;
}
if (but_c_flag == 1) {
    if (pwm_counter == 55) {
        if (!(PINA & (1 << STR1))) {
            pressed_button=13;
        }
        else if (!(PINA & (1 << STR2))) {
            pressed_button=11;
        }
        else if (!(PINA & (1 << STR3))) {
            pressed_button=0;
        }
        else if (!(PINA & (1 << STR4))) {
            pressed_button=12;
        }
    }
    else if (pwm_counter == 75) {
        if (!(PINA & (1 << STR1))) {
            pressed_button=14;
        }
        else if (!(PINA & (1 << STR2))) {
            pressed_button=3;
        }
        else if (!(PINA & (1 << STR3))) {
            pressed_button=2;
        }
        else if (!(PINA & (1 << STR4))) {
            pressed_button=1;
        }
    }
    else if (pwm_counter == 76) {
        if (!(PINA & (1 << STR1))) {
            pressed_button=15;
        }
        else if (!(PINA & (1 << STR2))) {
            pressed_button=6;
        }
        else if (!(PINA & (1 << STR3))) {
            pressed_button=5;
        }
        else if (!(PINA & (1 << STR4))) {
            pressed_button=4;
        }
    }
    else if (pwm_counter == 77) {
        if (!(PINA & (1 << STR1))) {
            pressed_button=16;
        }
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 123  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

        else if (!(PINA & (1 << STR2))) {
            pressed_button=9;
        }
        else if (!(PINA & (1 << STR3))) {
            pressed_button=8;
        }
        else if (!(PINA & (1 << STR4))) {
            pressed_button=7;
        }
    }
    but_c_flag=0;
}
if (pressed_button<17)
{
    TIMSK &= ~(1 << OCIE1A);
}
}

void timer1_init_key() {
    TCCR1A = 0;
    TCCR1B = 0;
    TCCR1B = (1 << WGM12) | (1 << CS11);
    OCR1A = 5;

    TIMSK |= (1 << OCIE1A);
    sei();
}

void process_digit_input() {

    if (pressed_button==14)
    {
        if (field_flag<3)
        {
            field_flag+=1;
        }else{
            field_flag=1;
        }
    }

    if (pressed_button==13)
    {
        if (field_flag>1)
        {
            field_flag-=1;
        }else{
            field_flag=3;
        }
    }

    if (field_flag==3)
    {
        //volume_now=0;
        if (pressed_button <= 9) {
            if (cost < 99999999) {
                cost = cost * 10 + pressed_button;
                volume_now=(cost*100)/price;
            }
        }
        else if (pressed_button == 12) {
            cost = 0;
            volume_now = 0;
        }
    }

    if(field_flag==2){

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 124  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |



```

        //cost=0;
        if (pressed_button <= 9) {
            if (volume_now < 999999999) {
                volume_now = volume_now * 10 + pressed_button;
                cost = (volume_now * price) / 100;
            }
        }
        else if (pressed_button == 12) {
            cost = 0;
            volume_now = 0;
        }
    }
    if(field_flag==1){
        if (pressed_button <= 9) {
            if (price < 9999999) {
                price = price * 10 + pressed_button;
            }
        }
        else if (pressed_button == 12) {
            price = 0;
        }
    }

    if (pressed_button == 11) {
        set_buzzer_mode(2);
        _delay_ms(150);
        refueling();
    }
    if (pressed_button != 17) {
        set_buzzer_mode(1);
        _delay_ms(400);
        pressed_button=17;
    }
}

void pass(){
    if(!(PINA & (1 << Dallas))){
        dallas_flag=1;
    }
    if(PINA & (1 << Dallas)){
        dallas_flag=0;
    }
}

void menu(){

    if (pressed_button==14)
    {
        menu_num+=1;
        if(menu_num>22){
            menu_num=1;
        }
    }

    if (pressed_button==13)
    {
        menu_num-=1;
        if(menu_num<1){
            menu_num=22;
        }
    }
    if (pressed_button==15)
    {
        menu_muv=0;
    }
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 125  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

if (pressed_button==16)
{
    menu_muv=2;
}
TIMSK &= ~(1 << OCIE1A);
if (menu_num==1)
{

    menu_ini1();
    process_menu0(menu1);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==2)
{
    menu_ini2();
    process_menu0(menu2);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==3)
{
    menu_ini3();
    process_menu0(menu3);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==4)
{
    menu_ini4();
    process_menu0(menu4);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==5)
{
    menu_ini5();
    process_menu0(menu5);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==6)
{
    menu_ini6();
    process_menu0(menu6);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==7)
{
    menu_ini7();
    process_menu0(menu7);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);

}
if (menu_num==8)

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 126  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

{
    menu_ini8();
    process_menu0(menu8);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==9)
{
    menu_ini9();
    process_menu0(menu9);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==10)
{
    menu_ini10();
    process_menu0(menu10);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==11)
{
    menu_ini11();
    process_menu0(menu11);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==12)
{
    menu_ini12();
    process_menu0(menu12);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==13)
{
    menu_ini13();
    process_menu0(menu13);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==14)
{
    menu_ini14();
    process_menu0(menu14);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==15)
{
    menu_ini15();
    process_menu0(menu15);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
}
if (menu_num==16)
{

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 127  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

        menu_ini16();
        process_menu0(menu16);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==17)
    {
        menu_ini17();
        process_menu0(menu17);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==18)
    {
        menu_ini18();
        process_menu0(menu18);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==19)
    {
        menu_ini19();
        process_menu0(menu19);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==20)
    {
        menu_ini20();
        process_menu0(menu20);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==21)
    {
        menu_ini21();
        process_menu0(menu21);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    if (menu_num==22)
    {
        menu_ini22();
        process_menu0(menu22);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);

    }
    set_buzzer_mode(1);
    pressed_button=17;

}

void refueling() {

    send_digit22(8);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    _delay_ms(800);
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 128  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

    set_buzzer_mode(3);
    _delay_ms(1000);
    litres_ini();
    TCNT1 = 0;
    timer1 = 0;
    cost=0;
    volume_old=volume_now;
    volume_now = 0;
    parse_number_to_digits(cost, volume_now, price, number) ;
    process_digits1(number);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    set_buzzer_mode(3);
    _delay_ms(1000);
    PORTD &= ~(1 << MOT);
    PORTC &= ~(1 << VCUT);
    PORTG &= ~(1 << VLOW);
    while (volume_now < volume_old) {
        timer1 = TCNT1;
        volume_now = timer1;
        cost = (volume_now * price) / 100;
        if ((volume_old - volume_now) < SLOWDOWN_THRESHOLD) {
            PORTC |= (1 << VCUT);
            PORTG &= ~(1 << VLOW);
        }
        parse_number_to_digits(cost, volume_now, price, number);
        process_digits1(number);
        PORTG |= (1 << PG2);
        PORTG &= ~(1 << PG2);
    }
    PORTD |= (1 << MOT);
    PORTC |= (1 << VCUT);
    PORTG |= (1 << VLOW);
}

void buzzer_timer2_init() {
    TCCR2=0;
    TCCR2 = (1 << CS22) | (1 << CS20);
    TIMSK |= (1 << TOIE2);
    sei();
}

void set_buzzer_mode(uint8_t mode) {
    buz_flag = mode;
    buz_state = 0;
    buz_counter = 0;
    pulse_count = 0;

    if (mode == 0) {
        TIMSK &= ~(1 << TOIE2);
        TCCR2=0;
        PORTA &= ~(1 << BUZ);
    } else {
        buzzer_timer2_init();
        PORTA |= (1 << BUZ);
    }
}

ISR(TIMER2_OVF_vect) {
    if (buz_flag == 0)
        return;
    buz_counter++;
    switch(buz_flag) {
        case 1:
            if (buz_counter >= TIME_50MS) {

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 129  |
| Змн. | Арк. | № документи | Підпис | Дата |                        |      |

```

        PORTA &= ~(1 << BUZ);
        buz_flag = 0;
    }
    break;
case 2:
    switch(buz_state) {
        case 0:
            if (buz_counter >= TIME_150MS) {
                PORTA &= ~(1 << BUZ);
                buz_state = 1;
                buz_counter = 0;
            }
            break;

        case 1:
            if (buz_counter >= TIME_50MS) {
                PORTA |= (1 << BUZ);
                buz_state = 2;
                buz_counter = 0;
            }
            break;
        case 2:
            if (buz_counter >= TIME_150MS) {
                PORTA &= ~(1 << BUZ);
                buz_flag = 0;
            }
            break;
    }
    break;
case 3:
    if (buz_counter >= TIME_150MS) {
        PORTA &= ~(1 << BUZ);
        buz_flag = 0;
    }
    break;
case 4:
    switch(buz_state) {
        case 0:
            if (buz_counter >= TIME_50MS) {
                PORTA &= ~(1 << BUZ);
                buz_state = 1;
                buz_counter = 0;
                pulse_count++;
            }
            break;
        case 1: // Пауза
            if (buz_counter >= TIME_100MS) {
                if (pulse_count < 4) {
                    PORTA |= (1 << BUZ);
                    buz_state = 0;
                    buz_counter = 0;
                } else {
                    buz_flag = 0;
                }
            }
            break;
    }
    break;
}

ISR(ANALOG_COMP_vect) {
    nozzle_flag = (ACSR & (1 << ACO)) ? 1 : 0;
}

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 130  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

int main() {
    DDRD|= (1 << MOT) ;
    DDRC|= (1 << VCUT) ;
    DDRG|= (1 << VLOW) ;
    PORTD|=(1 << MOT) ;
    PORTC|=(1 << VCUT) ;
    PORTG|=(1 << VLOW) ;
    DDRA |= (1 << DATA_PIN) | (1 << SCK_PIN) | (1 << BUZ);
    PORTA &= ~((1 << DATA_PIN) | (1 << SCK_PIN) | (1 << BUZ));
    DDRA &= ~((1 <<STR1 ) | (1 << STR2) | (1 <<STR3 )|(1 <<STR4 ));
    PORTA &= ~((1 <<STR1 ) | (1 << STR2) | (1 <<STR3 )|(1 <<STR4 ));
    DDRF |= (1 << KOL1) | (1 <<KOL2 ) | (1 << KOL3)| (1 <<KOL4 );
    PORTF &= ~((1 << KOL1) | (1 << KOL2) | (1 << KOL3)| (1 << KOL4 ));
    DDRG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    DDRA |= (1 << BUZ);
    PORTA &= ~(1 << BUZ);
    pass();
    AC_ini();
    start_paket();
    cost=0;
    volume_now=0;
    price=4232;
    parse_number_to_digits(cost, volume_now, price, number) ;
    process_digits1(number);
    PORTG |= (1 << PG2);
    PORTG &= ~(1 << PG2);
    set_buzzer_mode(3);
    _delay_ms(1000);
    while (1) {
        if (dallas_flag==0)
        {

            timer1_init_key();
            _delay_ms(200);
            menu();
            if (menu_num==0){refueling();}
            pass() ;
            }else{

            parse_number_to_digits(cost, volume_now, price,
number) ;

            process_digits1(number);
            PORTG |= (1 << PG2);
            PORTG &= ~(1 << PG2);
            pass() ;
            if (dallas_flag==1 &&nozzle_flag==1)
            {
                timer1_init_key();
                process_digit_input();
                parse_number_to_digits(cost, volume_now, price,
number) ;

                process_digits1(number);
                PORTG |= (1 << PG2);
                PORTG &= ~(1 << PG2);
            }
        }
    }
}

```

## ДОДАТОК Б

### Асемблері функції для індикації

```
.global process_digits1
process_digits1:
    push r16
    push r17
    push r28
    push r29
    push r19
    push r18
    ldi r18,0x00
    ldi r19,0x80
    movw r28, r24
    ldi r16, 22
    ;3,9,17
process_loop:
    ld r17, Y+
    cpi r17, 10
    breq call_n
    cpi r16, 4
    breq call_p
    cpi r16, 12
    breq call_p
    cpi r16, 20
    breq call_p
    rjmp call_n
call_n:
    call call_num
    rjmp next_digit

call_p:
    call call_point
    rjmp next_digit

next_digit:
    dec r16
    brne process_loop
    call sck_3_pak
    pop r19
    pop r18
    pop r29
    pop r28
    pop r17
    pop r16
    ret

.global call_num
call_num:
    cpi r17, 0
    breq call_send_0
    cpi r17, 1
    breq call_send_1
    cpi r17, 2
    breq call_send_2
    cpi r17, 3
    breq call_send_3
    cpi r17, 4
    breq call_send_4
```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 132  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |



```

        cpi r17, 5
        breq call_send_5
        cpi r17, 6
        breq call_send_6
        cpi r17, 7
        breq call_send_7
        cpi r17, 8
        breq call_send_8
        cpi r17, 9
        breq call_send_9
        cpi r17, 10
        breq call_send_10
        ret
call_send_0:
    call send_0
    ret
call_send_1:
    call send_1
    ret
call_send_2:
    call send_2
    ret
call_send_3:
    call send_3
    ret
call_send_4:
    call send_4
    ret
call_send_5:
    call send_5
    ret
call_send_6:
    call send_6
    ret
call_send_7:
    call send_7
    ret
call_send_8:
    call send_8
    ret
call_send_9:
    call send_9
    ret
call_send_10:
    call send_10
    ret

.global call_point

call_point:
    cpi r17, 0
    breq call_send0_point
    cpi r17, 1
    breq call_send1_point
    cpi r17, 2
    breq call_send2_point
    cpi r17, 3
    breq call_send3_point
    cpi r17, 4
    breq call_send4_point
    cpi r17, 5
    breq call_send5_point
    cpi r17, 6

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 133  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

        breq call_send6_point
        cpi r17, 7
        breq call_send7_point
        cpi r17, 8
        breq call_send8_point
        cpi r17, 9
        breq call_send9_point
        ret
call_send0_point:
    call send_0_point
    ret
call_send1_point:
    call send_1_point
    ret
call_send2_point:
    call send_2_point
    ret
call_send3_point:
    call send_3_point
    ret
call_send4_point:
    call send_4_point
    ret
call_send5_point:
    call send_5_point
    ret
call_send6_point:
    call send_6_point
    ret
call_send7_point:
    call send_7_point
    ret
call_send8_point:
    call send_8_point
    ret
call_send9_point:
    call send_9_point
    ret

.global sck_3_pak

sck_3_pak:
    out 0x1B, r19
    nop
    nop
    nop
    nop
    out 0x1B, r18
    out 0x1B, r19 ;2
    nop
    out 0x1B, r18
    out 0x1B, r19
    nop
    out 0x1B, r18
    out 0x1B, r19
    nop
    out 0x1B, r18
    out 0x1B, r19
    nop
    out 0x1B, r18
    out 0x1B, r19
    nop
    out 0x1B, r18

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 134  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
nop
nop
nop
nop
out 0x1B, r19
nop
nop
nop
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop
out 0x1B, r18
out 0x1B, r19
nop

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 135  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |

```

out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
nop
nop
nop
nop
out 0x1B, r19
nop
nop
nop
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B,r19
nop
out 0x1B,r18
out 0x1B, r19
clr r18
out 0x1B,r18
ret

```

|      |      |             |        |      |                        |      |
|------|------|-------------|--------|------|------------------------|------|
|      |      |             |        |      | 171.4321.КР.ПЗ.Додатки | Арк. |
|      |      |             |        |      |                        | 136  |
| Змн. | Арк. | № документа | Підпис | Дата |                        |      |