

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: Комп'ютеризована система управління силовими розетками

Виконав: студент IV курсу, групи СІ-41

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Головатий А. Р.

(прізвище та ініціали)

Керівник

(підпис)

Жаровський Р.О.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тим С.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

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

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

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

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Осухівська Г.М.  
(підпис) (прізвище та ініціали)  
«\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

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

студента Головатого Артура Романовича  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система управління силовими розетками

Керівник роботи Жаровський Руслан Олегович, к.т.н., доцент кафедри КС  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» квітня 2024 року № 4/7-408.

2. Термін подання студентом завершеної роботи 24.06.2024 р.

3. Вихідні дані до роботи Технічне завдання

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

Вступ. 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

Функціональна схема системи.

Схема електрична принципова.

Макет і локальний інтерфейс

Веб-інтерфейс налаштування системи

### 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., професор кафедри МТ</i>		

7. Дата видачі завдання

25.04.2024

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>01.02-09.02.2024</i>	
2	<i>Аналіз технічного завдання</i>	<i>05.02-11.02.2024</i>	
3	<i>Визначення і аналіз вимог до комп'ютеризованої системи</i>	<i>25.04-03.05.2024</i>	
4	<i>Проектування структури комп'ютеризованої системи</i>	<i>04.05-20.05.2024</i>	
5	<i>Проектування програмної частини</i>	<i>21.05-28.05.2024</i>	
6	<i>Проектування апаратної частини</i>	<i>29.05-06.06.2024</i>	
7	<i>Налаштування і тестування комп'ютеризованої системи</i>	<i>07.06-10.06.2024</i>	
8	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>10.06-15.06.2024</i>	
9	<i>Оформлення кваліфікаційної роботи</i>	<i>16.06-20.06.2024</i>	
10	<i>Попередній захист кваліфікаційної роботи</i>	<i>14.06.2024</i>	
11	<i>Захист кваліфікаційної роботи</i>	<i>24.06-28.06.2024</i>	

Студент

\_\_\_\_\_  
(підпис)

*Головатий Артур Романович*

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

*Жаровський Руслан Олегович*

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

Комп'ютеризована система управління силовими розетками // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Головатий Артур Романович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ-41 // Тернопіль, 2024 // с.– 70, рис. – 38, табл. – 1, аркушів А1 – 4, бібліогр. – 18.

Ключові слова: мікроконтроллер, керування, силові розетки.

У даній кваліфікаційній роботі виконано розробку програмно-апаратного комплексу для керування силовими розетками.

Пояснювальна записка містить 4 розділи.

В першому розділі здійснюється аналіз предметної області. Проведено огляд можливих варіантів реалізації системи управління, а також визначені задачі кваліфікаційної роботи.

В другому розділі описані компоненти і особливості проектування системи керування силовими розетками.

В третьому розділі проведена реалізація налаштування і тестування системи керування силовими розетками.

В четвертому розділі розглянуті питання охорони праці та вимоги з техніки безпеки.

## ABSTRACT

Computerized control system for power outlets // Bachelor`s thesis // Golovatyy Artur Romanovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Systems and Networks, group CI-41 // Ternopil, 2024 // p.– 70, fig. - 38, table – 1, sheets A1 – 4, bibliography. - 18.

Keywords: microcontroller, control, power outlets.

In this qualification work, the software and hardware complex for controlling power outlets was developed.

The explanatory note contains 4 sections.

In the first section, the subject area is analyzed. An overview of possible variants of the implementation of the control system was carried out, as well as the tasks of the qualification work were defined.

The second section describes the components and design features of the power outlet control system.

In the third section, the configuration and testing of the power outlet control system was implemented.

The fourth chapter deals with occupational health and safety requirements.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ .....	10
1.1 Огляд існуючих рішень .....	10
1.1.1 Універсальне електромагнітне реле.....	10
1.1.2 Оптопара .....	12
1.1.3 Транзистор .....	13
1.1.4 Твердотільне реле (SSR DC).....	14
1.1.5 Симістор як вкл/викл.....	14
1.2 Вимоги до системи керування силовими розетками.....	15
1.3 Постановка задач кваліфікаційної роботи.....	17
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА.....	18
2.1 Розробка структури системи управління силовими розетками .....	18
2.2 Обґрунтування вибору апаратного забезпечення.....	19
2.2.1 Розробка принципової схеми пристрою .....	26
2.3 Обґрунтування вибору програмного забезпечення.....	31
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	34
3.1 Реалізація та моделювання проектних рішень .....	34
3.1.1 Реалізація апаратної складової комп'ютеризованої системи.....	34
3.1.2 Реалізація програмної складової системи .....	37
3.2 Тестування системи .....	39

					КС КРБ 123.115.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.	Головатий А.				Літ.	Арк.	Аркуші
Перевір.	Жаровський				6		
Реценз.					ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.	Тиш Є.В.						
Затверд.	Осухівська Г.М.						
					Комп'ютеризована система управління силовими розетками		

РОЗДІЛ 4	БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..	45
4.1	Ергономічні проблеми безпеки життєдіяльності .....	45
4.2	Вимоги пожежної безпеки при гасінні електроустановок .....	48
	ВИСНОВКИ.....	50
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
	Додаток А. Технічне завдання .....	53
	Додаток Б. Лістинг програмного забезпечення .....	62

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Сучасний світ не може обійтися без автоматики, що полегшує людську працю як у побуті, і на виробництві. Також проблема енергозбереження в останні роки набула особливої актуальності. Організації і побутові користувачі задіюють величезні зусилля щодо впровадження інноваційних рішень та технологій, що дозволяють скоротити споживання електроенергії.

Гострота проблеми енергопостачання пов'язана з двома обставинами: високі тарифи на електроенергію та дефіцит електроенергії в останні роки. На перший чинник організації і побутові користувачі мало впливають, то другий значною мірою визначається адміністрацією організації або користувачем. Проблема низької енергоефективності особливо актуальна великих вишів з далеко розкиданими об'єктами.

Низька енергоефективність має наступні причини:

- недбалість та забудькуватість у відключенні освітлення, електроприладів, обладнання та побутових приладів;
- відсутність стимулів економії в організаціях;
- не використання енергозберігаючих технологій.

Серед способів скорочення витрати електроенергії найбільш ефективним застосування автоматичних систем управління електроживленням. Існуючі системи електроживлення у переважній більшості випадків не задовольняють вимогам енергозбереження, що веде до збільшення витрат на електроенергію. У той же час за рахунок використання систем керування електроживленням можна суттєво знизити витрати електроенергії.

Використання передових інформаційних технологій дозволяє автоматизувати процес управління. Багатофункціональні процесорні пристрої можна запрограмувати на виконання безлічі завдань, однією з яких є виконання періодичного включення і вимкнення різного роду корисних електричних приладів. Для вирішення цього завдання застосовуються таймери, що масово випускаються промисловістю, з різною

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		



функціональністю. Більшість таких приладів розраховані на добове або тижневе використання та, на жаль, керують лише одним або рідко двома каналами.

Беручи до уваги вище описане, у даній випускній кваліфікаційній роботі і буде приділено увагу розробці недорогої системи управління електроживлення.

Метою кваліфікаційної роботи є розробка комп'ютеризованої системи, яка забезпечить скорочення електроспоживання та оптимізує роботу електрообладнання.

					<i>КС КРБ 123.115.00.00 ПЗ</i>	Арк.
						9
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Огляд існуючих рішень

Досягнення сучасної електроніки дозволили створити різноманітні пристрої для безпосереднього керування силовими розетками на базі мікроконтролерів. Тому спершу розглянемо які існують методи і апаратні засоби керування силовими розетками.

#### 1.1.1 Універсальне електромагнітне реле

Електромагнітне реле є, по суті, керованим механічним вимикачем: подали на нього струм – воно замкнуло контакти, зняли струм – розімкнуло. Контакти є саме контактами: металевими "п'ятаками", що притискаються один до одного. Саме тому таке реле може керувати як навантаженням постійного, і змінного струму.

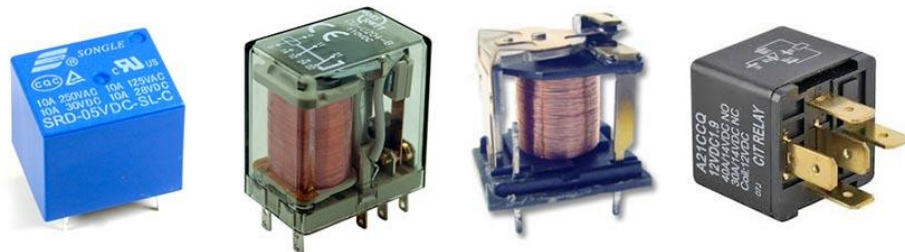


Рисунок 1.1 - Електромагнітне реле

Сама котушка реле є індуктивним навантаженням, що призводить до додаткових проблем, тому для управління потрібен додатковий силовий та захисний ланцюг.

					<b>КС КРБ 123.115.00.00 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Головатий А.			Аналіз технічного завдання	Літ.	Арк.	Аркушів
Перевір.		Жаровський					10	
Реценз.						ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

Такі модулі бувають як з одним так і з кількома реле на платі.

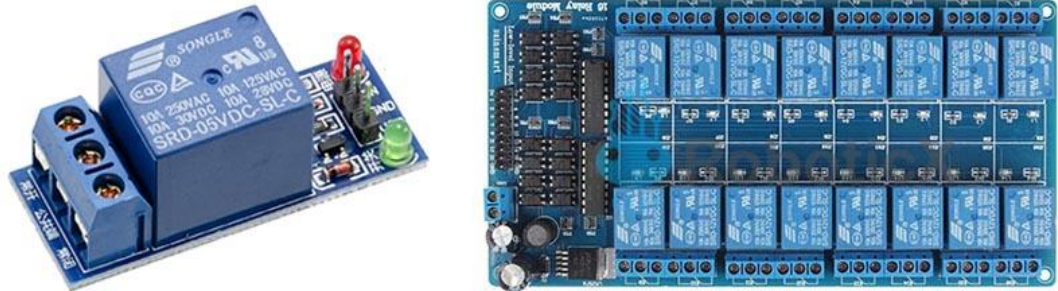


Рисунок 1.2 – Варіанти реалізації модулів з реле

Таке реле зроблено спеціально для зручного керування з мікроконтролера: пін живлення VCC ( $V_{in}$ , 5V) та GND підключаються до живлення, а далі реле керується логічним сигналом, поданим на пін IN . З іншого боку стоїть клемник для підключення проводів, зазвичай контакти підписані як NO , NC і COM .

Такі модулі реле бувають двох типів: низького та високого рівня. Реле низького рівня перемикається за наявності низького сигналу (GND) на керуючому пині digitalWrite ( pin, LOW ). Реле високого рівня відповідно спрацьовує від високого рівня digitalWrite ( pin, HIGH ). Також існують модулі з вибором рівня:

Електромагнітне реле має ряд недоліків перед іншими розглянутими нижче способами:

- обмежена кількість перемикачів: механічний контакт зношується, особливо при великому та/або індуктивному навантаженні;
- голосно спрацьовує;
- при великому навантаженні реле може "залипнути", тому для великих струмів потрібно використовувати більш потужні реле, які доведеться включати за допомогою маленьких реле, або транзисторів.

- необхідні додаткові ланцюги для керування реле;
- дуже великі наведення на всю лінію живлення під час комутації індуктивного навантаження;
- відносно довге перемикання, при управлінні індуктивними ланцюгами змінного струму можна отримати великий індуктивний викид, необхідно ставити іскрогасні ланцюги.

### 1.1.2 Оптопара

Оптопара – відмінний елемент, що дозволяє виконувати дві функції: комутувати навантаження (нехай і невелике) і повністю фізично розв'язує мікроконтролер з нею. Оптопари можна використовувати для імітації натискання кнопок інших зовнішніх пристроїв, тобто замикати суто логічний сигнал. Також можна використовувати для розривання живлення різних датчиків та модулів у пристрої замість транзистора. Оптопара складається з двох частин: світлодіод, який ми включаємо за допомогою мікроконтролера, і вихідна частина, яка може бути різною (транзистор, симистор та ін.), таким чином сигнал з мікроконтролера відокремлюється від навантаження через промінь світла, що дуже важливо при комутації високовольтних або якихось чутливих кіл. Для управління зовнішніми пристроями треба брати оптопари з транзисторним виходом, наприклад дуже поширену РС814 та її аналоги (рис.1.3).

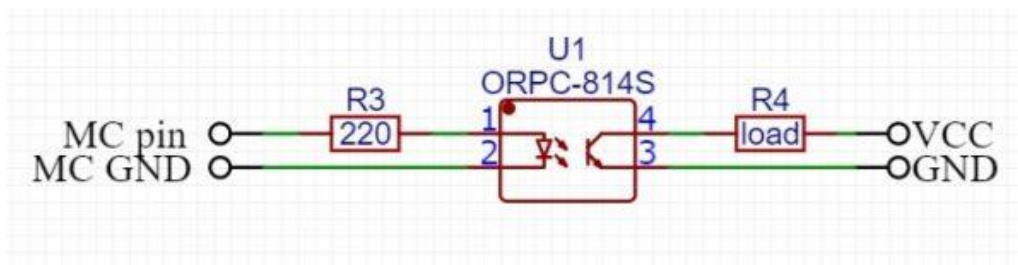


Рисунок 1.3 – Підключення оптопари

Підключається оптопара у такий спосіб (рис.1.3): світлодіодом ми керуємо з МК через резистор, а вихід підключаємо в розрив навантаження, дотримуючись полярності.

Зазначу, що оптопара не призначена для управління великим навантаженням, зазвичай це комутація інших логічних ланцюгів

### 1.1.3 Транзистор

Найкомпактніший спосіб керувати навантаженням постійного струму – транзистор. Транзистори бувають біполярні та польові (MOSFET, ключ). Біполярні вже морально та фізично застаріли, мають багато характеристик та потребують додаткового вивчення теми, тому ми розглянемо лише польові транзистори. Схема типова і виглядає так (рис.1.4):

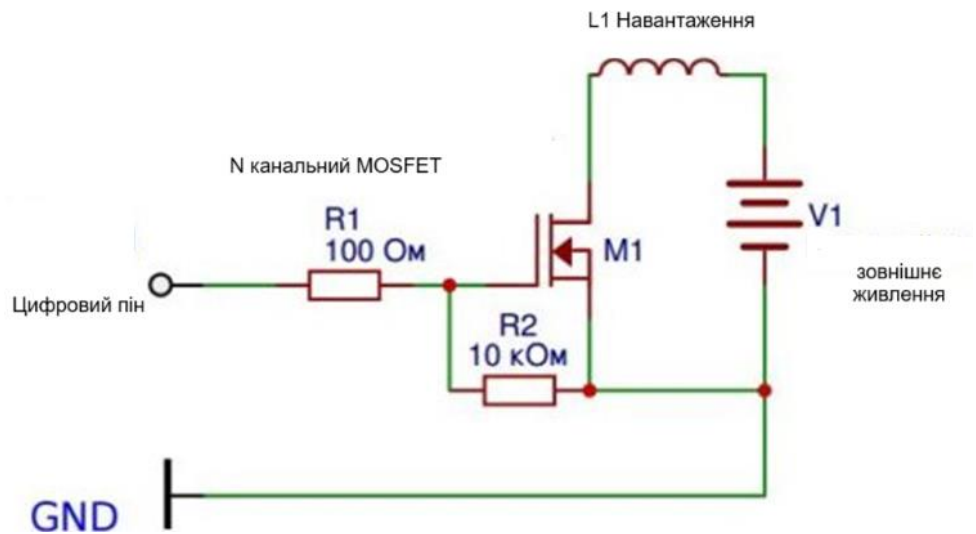


Рисунок 1.4 - Схема підключення транзистора

Найважливіший момент: на польовий транзистор можна подавати ШІМ сигнал для "плавного" керування навантаженням: плавно змінювати швидкість обертання мотора, яскравість світлодіодної стрічки, потужність обігрівача та інше!

### 1.1.4 Твердотільне реле (SSR DC)

Твердотільне реле (рис.1.5) підключається безпосередньо до Arduino, пін “-” до GND, “+” до будь-якого цифрового піну. Вихід реле ставиться у розрив ланцюга живлення навантаження, як вимикач. Важливо не переплутати плюс і мінус, тому що всередині реле є польовий транзистор на радіаторі



Рисунок 1.5 - Система управління навантаженням на базі твердотільного реле

### 1.1.5 Симістор як вкл/викл

Симістор – це радіоелемент, схожий на транзистор, але може працювати на змінному струмі. Для керування симістором використовується оптопара із симісторним виходом. Найпростіша схема підключення наведена на рис. 1.6.

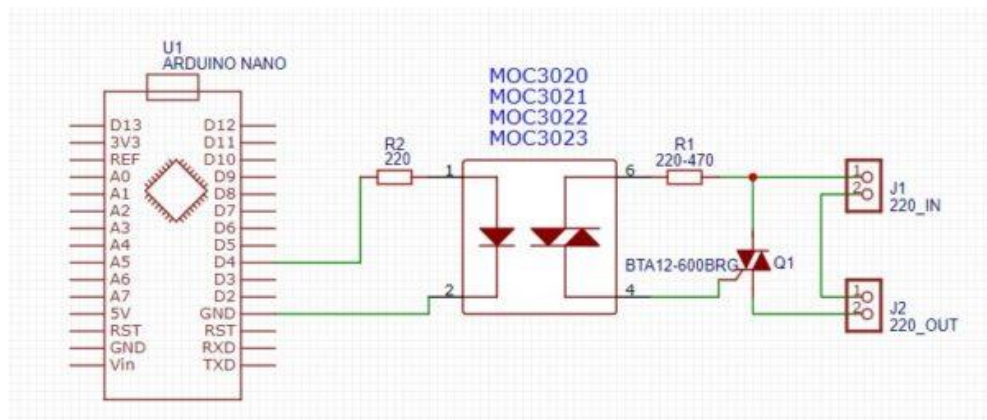


Рисунок 1.6 – Підключення симістора



Для керування навантаженням тільки в режимі вкл/викл бажано ставити оптопару з детектором нуля (наприклад МОС306х ), вона сама відключатиме і включатиме навантаження тільки в моменти переходу напруги в мережі через 0, що сильно зменшує перешкоди в мережі. Також тут стоять резистори: 220 Ом – для обмеження струму на світлодіод оптопари. І резистор між оптопарою і симістором: 220-470 Ом з потужністю 1-2 Вт. Симістор потрібно брати з хорошим запасом струму, щоб менше грівся.

Симістор є більш універсальним і може використовуватись як димер. Для цієї схеми потрібна оптопара без детектора нуля, наприклад, серії МОС302х . Схема такого виробу наведена на рис. 1.7.

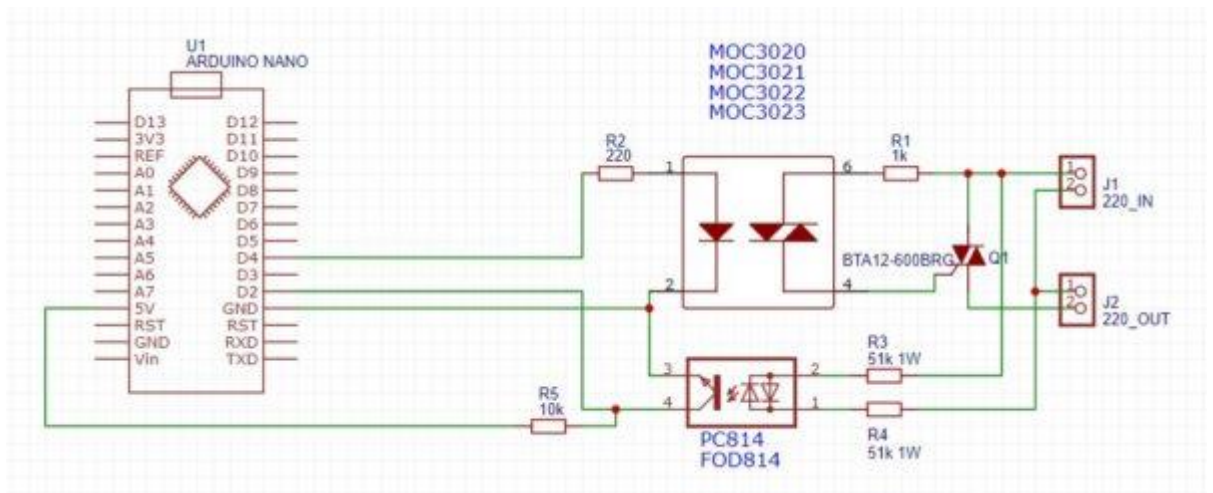


Рисунок 1.7 – Підключення симістора для реалізації диммера

Провівши такий огляд більш універсальною є останній варіант, оскільки він дозволяє керувати також освітленням в якості диммера.

## 1.2 Вимоги до системи керування силовими розетками

Відповідно до технічного завдання, система має містити централізовану систему управління через канал зв'язку Ethernet контролерів та сервера. Контролери здійснюють керування навантаженням.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Сервер формує завдання для контролерів та організує взаємодію операторів із системою через web інтерфейс з будь-якого комп'ютера чи мобільного пристрою, приєднаного до мережі Інтернет.

Необхідно розробити мікропроцесорний електронний контроллер із встановленням часу спрацьовування від 1 хвилини до 24 годин з періодичністю включення таймера від 1 доби до 30 діб. Виходячи з необхідної кількості імпульсів, що вважаються, в характерних інтервалах:

- 1 хв. = 60 с. \* 1000 мс = 60 000 мс;
- 1 добу. = 24 год. \* 60 хв. = 1440 хв.;
- 30 діб. = 1440 хв. \* 30 = 43200 хв.

Це менше  $2^{16} = 65536$ , тому будемо використовувати шістнадцятибітний таймер.

Таким чином, структура пристрою, що розробляється, повинна складатися з наступних блоків:

- мікроконтролер (МК);
- панель програмування та індикатор МК;
- блок комутації навантажень;
- блок живлення АС 220В, 50Гц/ DC 5В.

Для функціонування 4-х каналного електронного таймера необхідно написати програму ( sketch ), яка виконує такі функції:

- вибір каналу керування;
- встановлення часу вмикання таймера в межах доби;
- встановлення часу вимкнення таймера в межах доби;
- встановлення періодичності повторення таймера;
- збереження налаштувань таймера в енергонезалежну пам'ять;
- перехід з режиму налаштування до режиму індикації після закінчення таймера індикації автоматично або за командою з панелі керування вручну;

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16



- індикація стану всіх таймерів на одному екрані.

Розроблений Sketch записується в постійну пам'ять МК через відповідні програмні засоби ( IDE ) з ПК, відповідно до інструкції з програмування МК.

### 1.3 Постановка задач кваліфікаційної роботи

На основі проведеного огляду та аналізу можливих рішень, а також з урахуванням наведених вимог, мета кваліфікаційної роботи буде досягнута шляхом виконання таких завдань:

- розробка структури системи управління силовими розетками;
- обґрунтування вибору модулів і розробка апаратної частини;
- обґрунтування і розробка програмного забезпечення;
- виготовлення, складання та налаштування макетного зразка системи управління силовими розетками;
- тестування системи управління силовими розетками.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

## РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Розробка структури системи управління силовими розетками

Для реалізації системи управління силовими розетками необхідний наступний набір підсистем:

- клієнтська програма, яка буде надавати доступ до системи керування з ПК, планшету або мобільного телефону;
- графічний інтерфейс програми через веб-форму для універсальності підключення з різних пристроїв;
- база даних, яка забезпечує зберігання даних користувачів;
- ядро системи, яке представляє собою мікроконтроллер, плата яка здійснює керування силовими розетками, датчики;
- система локального контролю, яка дозволяє з допомогою кнопок і дисплею безпосередньо керувати параметрами електроживлення;
- програмне забезпечення мікроконтроллера.

Розробимо спершу функціональну схему системи керування силовими розетками. На даній схемі розмістимо вищезгадані компоненти і відповідні зв'язки між ними. На рис. 2.1 представлено функціональну схему системи.

Як видно з даного рисунку система ядром системи буде мікроконтроллер з відповідним програмним забезпеченням. Для його налаштування можна використовувати локальну клавіатуру і екрана також можна скористатись зручним веб інтерфейсом. Веб інтерфейс дозволяє здійснювати підключення віддалено і здійснювати налаштування параметрів керування силовими розетками з любого пристрою на якому встановлено веб-браузер і є доступ до мережі Інтернет.

					<b>КС КРБ 123.115.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Головатий А.</i>			<i>Проектна частина</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Жаровський</i>					18	
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Для опису структури і можливих шляхів взаємодії компонентів системи необхідно побудувати її структурну схему.

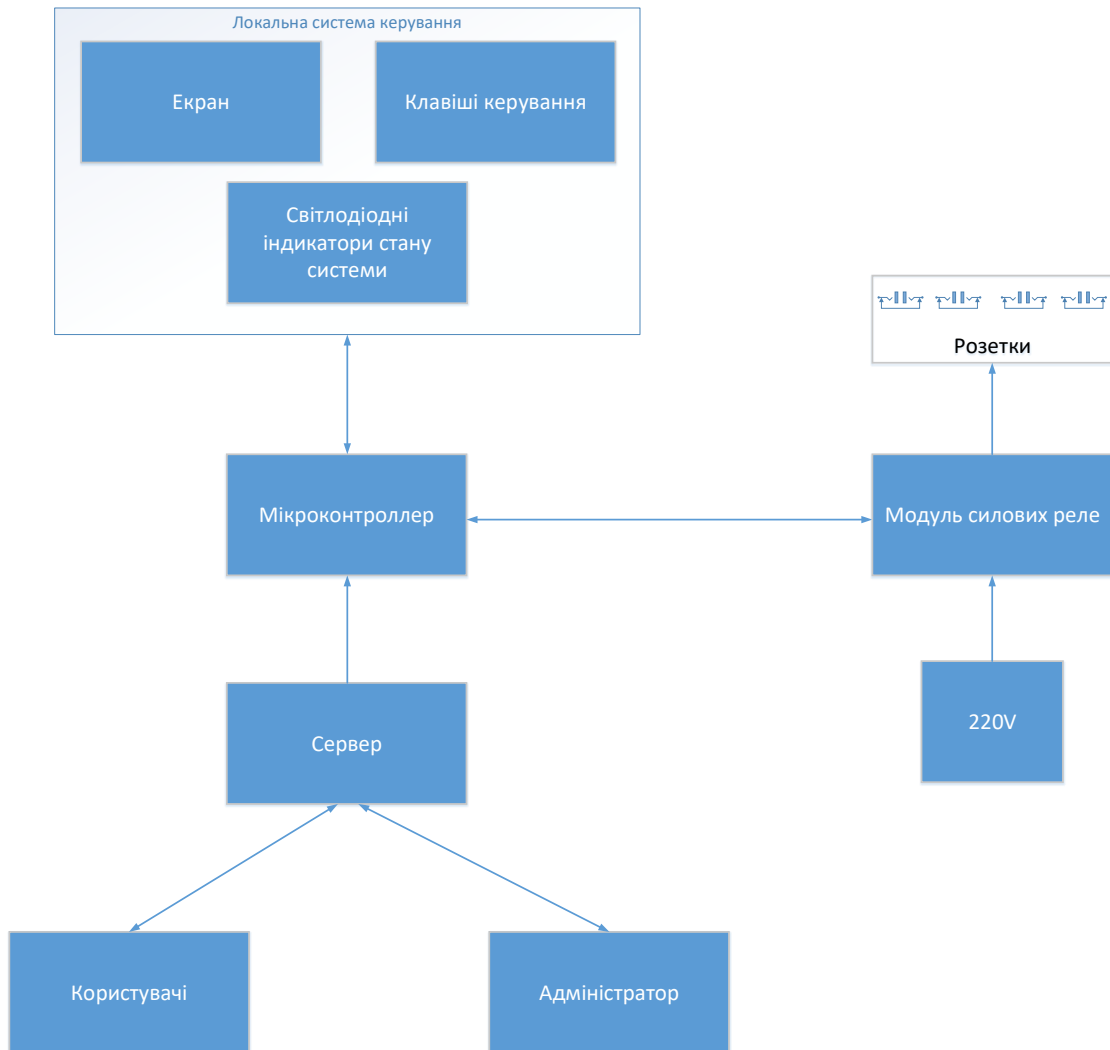


Рисунок 2.1 – Функціональна схема системи

Для розробки пристрою необхідно визначитися з технологіями, що будуть використовуватись для системи, а також необхідно здійснити вибір апаратного і програмного забезпечення.

## 2.2 Обґрунтування вибору апаратного забезпечення

Головним про проектуванні любого пристрою є його універсальність і можливість легкої реалізації. При виборі елементної бази будемо старатись

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

використовувати універсальні компоненти, які легко замовити і ціна таких компонент, на відміну від спеціалізованих, буде невеликою.

Так як ми визначили що головним ядром системи є мікроконтроллер то вибір апаратного забезпечення розпочнемо з нього. Розглянемо деякі варіанти апаратних обчислювальних платформ, що підходять для виконання поставленого завдання та порівняємо їх характеристики.

Різновидів мікроконтроллерів і варіантів їх реалізації є значна кількість. В таблиці 2.1 наведу базові технічні характеристики найбільш поширених.

Таблиця 2.1 – Порівняння характеристик апаратних обчислювальних платформ

Назва	Arduino Uno	Arduino Nano	Arduino Mega 2560	Arduino Leonardo
Мікроконтролер	ATmega328	Atmel ATmega168 ATmega328	ATmega2560	ATmega32u4
Робоча напруга, В	5	5	5	5
Рекомендована напруга живлення, В	7 – 12	7 – 12	7 – 12	7 – 12
Максимальна напруга живлення, В	6 – 20	6 – 20	6 – 20	6 – 20
Цифрові входи/виходи	14 (з них 6 можуть використовуватися як ШІМ-виходів)	14 (з них 6 можуть використовуватися як ШІМ-виходів)	54 (з них 14 можуть використовуватися як ШІМ-виходів)	20 (з них 7 можуть використовуватися як ШІМ-виходів))
Аналогові входи	6	6	16	12
Максимальний струм одного виходу, мА	40	40	40	40
Максимальний вихідний струм виходу 3.3, мА	50	50	50	50

Назва	Arduino Uno	Arduino Nano	Arduino Mega 2560	Arduino Leonardo
Flash-пам'ять, КБ	32	16 /32	256	32
SRAM, КБ	2	2	8	2
EEPROM, КБ	1	0,5	4	1
Тактова частота, МГц	16	16	16	16

Як МК обраний Arduino MEGA 2560 [2], що має у своєму складі 6 програмованих таймерів:

- восьми бітних (макс. значення, що відраховує 256);
- 4 шістнадцяти бітних (макс. значення, що відраховується 65 536).

Інші плати Arduino реалізовані на основі 8-ми бітного процесора, містять у своєму складі 8-ми бітний таймер, за винятком плати Duo, що має 32-бітний процесор і вищу вартість [2].

Інший можливий варіант вирішення завдання – використання зовнішніх годинників реального часу (RTC) з кварцовою стабілізацією та живленням на батарейках, наприклад, DS 3231.

МК Arduino MEGA 2560 побудований на основі ATmega 2560 із тактовою частотою 16МГц [4]. МК має 54 порти вводу/виводу; 15 вихідних портів можуть працювати, як вихід з ШІМ сигналом (аналоговий вихід); 16 аналогових входів з АЦП можуть обробляти сигнали аналогових датчиків. Струм, що видається кожним портом у навантаження – не більше 40 мА [6].

Зовнішнє живлення 7÷20В зі зниженням через вбудований у плату стабілізатор 5В до виходу якого можливе підключення зовнішніх плат та датчиків із сумарним струмом не більше 800мА [8].

Для зберігання програми передбачена Flash пам'ять розміром 256 КБ (8 КБ з них використовуються завантажувачем); оперативна пам'ять ОЗП SRAM розміром 8 КБ; енергонезалежна пам'ять EEPROM розміром 4 КБ [10].

Для зв'язку з різними пристроями передбачено 4 UART інтерфейси, у ролі виступають висновки 0, 1, 14-19. Один із портів спрямований на USB через мікроконтролер ATmega16U2 [12].

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

На платі передбачені роз'єми для встановлення плат розширення Shield. Розміри плати: 53x110x12, 5мм. Роз'єм зовнішнього живлення під штекер діаметром 2,1 мм з плюсом посередині. Зовнішній вигляд плати з обох сторін наведено на рис. 2.2.

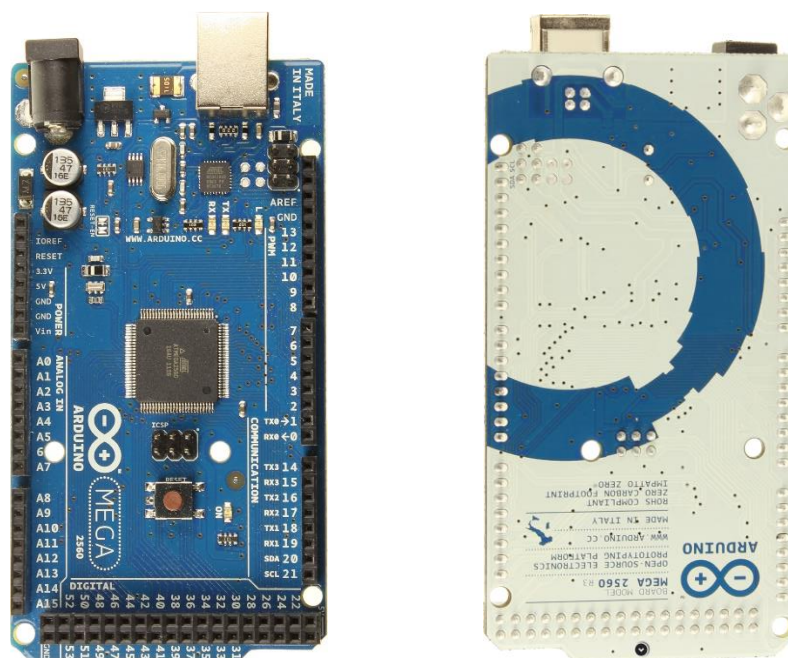


Рисунок 2.2 – Зовнішній вигляд плати Arduino MEGA 2560

Далі необхідно підібрати панель керування яка буде містити екран кнопки керування і індикатори. Панель керування пристроєм та індикатор МК, вибрані з урахуванням сумісності з МК Arduino MEGA 2560 та вартості реалізації, у вигляді готової плати розширення Arduino LCD Keypad Shield [13]. Зовнішній вигляд плати з обох сторін наведено на рис. 2.3. Схема плати з позначенням виводів наведена на рис. 2.4.



Рисунок 2.3 – Зовнішній вигляд плати розширення LCD Keypad МК

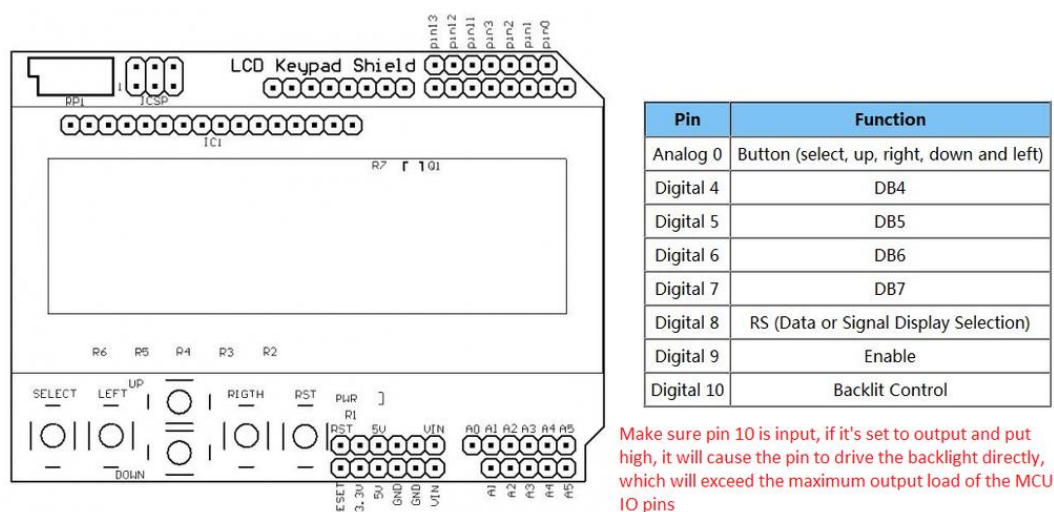


Рисунок 2.4 – Схема плати Arduino LCD Keypad Shield

Плата розширення LCD Keypad Shield містить: індикатор LCD 1602, що дозволяє відобразити два ряди по 16 символів, підключений до чотирьох цифрових виходів Arduino D 4- D 7; шість кнопок ( RIGHT, UP, DOWN, LEFT, SELECT ), підключених до аналогового входу Arduino A0; кнопка RESET, підключена до входу RST ; резистор налаштування контрастності дисплея LCD. Розміри плати 60x83x20мм.

Для підключення системи керування силовими розетками до мережі можна скористатись технологіями WiFi або Ethernet. Зважаючи, що даний



пристрій може бути встановлений в щитовій чи в інших місцях де проходить значна кількість силових ліній, то перевагу надамо Ethernet підключенню.

Як і в попередньому випадку використаємо готове рішення Ethernet модуль ENC28J60 (рис.2.5).

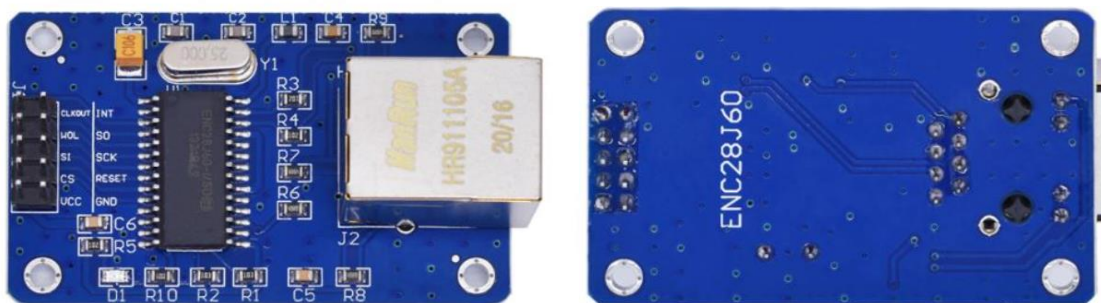


Рисунок 2.5 – Ethernet модуль ENC28J60

Модуль ENC28J60 підключається до Arduino через інтерфейс SPI, з тактовою частотою до 20 МГц. Для підключення до мережі TCP/IP використовується роз'єм RJ-45, при цьому модуль має гальванічну розв'язку. Контролер модуля працює на максимальній тактовій частоті 25 МГц і відповідає за всі функції роботи з мережними протоколами, тоді як Arduino лише відправляє та приймає пакети даних.

Модуль працює при напрузі 3,3 В, що робить його сумісним з мікроконтролерами STM32 або STM8. Максимальний споживаний струм може досягати 250 мА, а номінальний становить 170 мА.

Схема підключення ENC28J60 плати на базі Arduino MEGA 2560 виглядатиме так як показано на рис. 2.6. Призначення контактів наступне:

- VCC - 3v3;
- GND - GND;
- CS - 53;
- SI - 51;
- SO – 50;



– SCK - 52.

Всі плати пристрою живляться від 5В, при цьому струми споживання:

- МК Arduino MEGA2560 (CPU+I/O)  $I \approx 50\text{mA}$ ;
- LCD Keypad Shield (LCD)  $I \approx 1\text{mA} + 130\text{mA}$  (Back Light LED);
- плата перемикавання навантаження  $I \approx 20\text{mA} * 4 = 80\text{mA}$ .

Разом сумарно:  $I \text{ потр} \approx 260\text{mA}$ .

Виберемо блок живлення імпульсного типу зі стабілізацією AC 220В, 50Гц/ DC 9В, 600mA. Блок живлення стабілізований імпульсний з ЕМСфільтром на вході. В цьому випадку на стабілізаторі 5В виділятиметься потужність порядку 1Вт. [7] Можна зменшити потужність розсіювання до 0,5Вт, перенісши живлення Back Light LED на шину  $V_{IN} = 9\text{V}$ . Розміри плати: 51x24x19мм. Зовнішній вигляд плати та кабелю живлення плати МК наведено рис. 2.6.

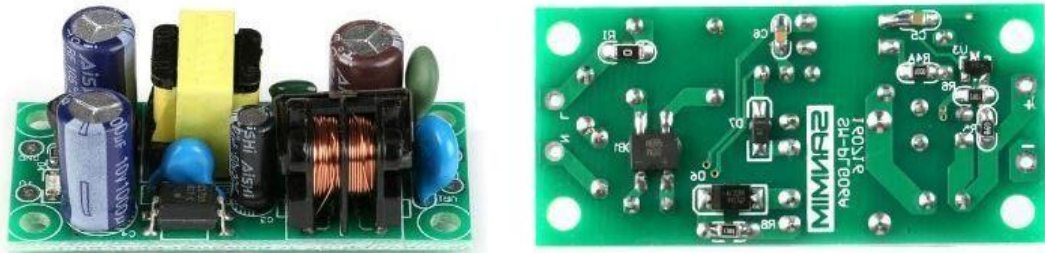


Рисунок 2.6 – Зовнішній вигляд плати блоку живлення

Металевий корпус із розетками для підключення навантажень розробляється для кріплення вертикально на стіну, для чого з боків корпусу передбачені куточки. На передній панелі розташовуються: розташовані ряд розетки для включення навантажень, нижче - екран LCD і кнопки управління електронним таймером. Передня панель знімна для сервісного обслуговування та доступу до USB МК. Регулювання контрастності LCD та кнопка RESET для скидання МК доступні через технологічні отвори діаметром 3 мм поруч із

екраном під викрутку. Для підключення загального живлення пристрою на бічній панелі корпусу пристрою розміщено роз'єм типу ІЕК 320- С 14 у колодці із запобіжником та загальним вимикачем живлення. Передня панель знімна для сервісного обслуговування та доступу до USB МК. [5]

### 2.2.1 Розробка принципової схеми пристрою

Розроблений пристрій збирається з готових плат тому принципова схема приладу буде складатись з окремих конструктивних елементів.

Принципова схема основної процесорної частини плати Arduino MEGA 2560 зображено на рис. 2.7.

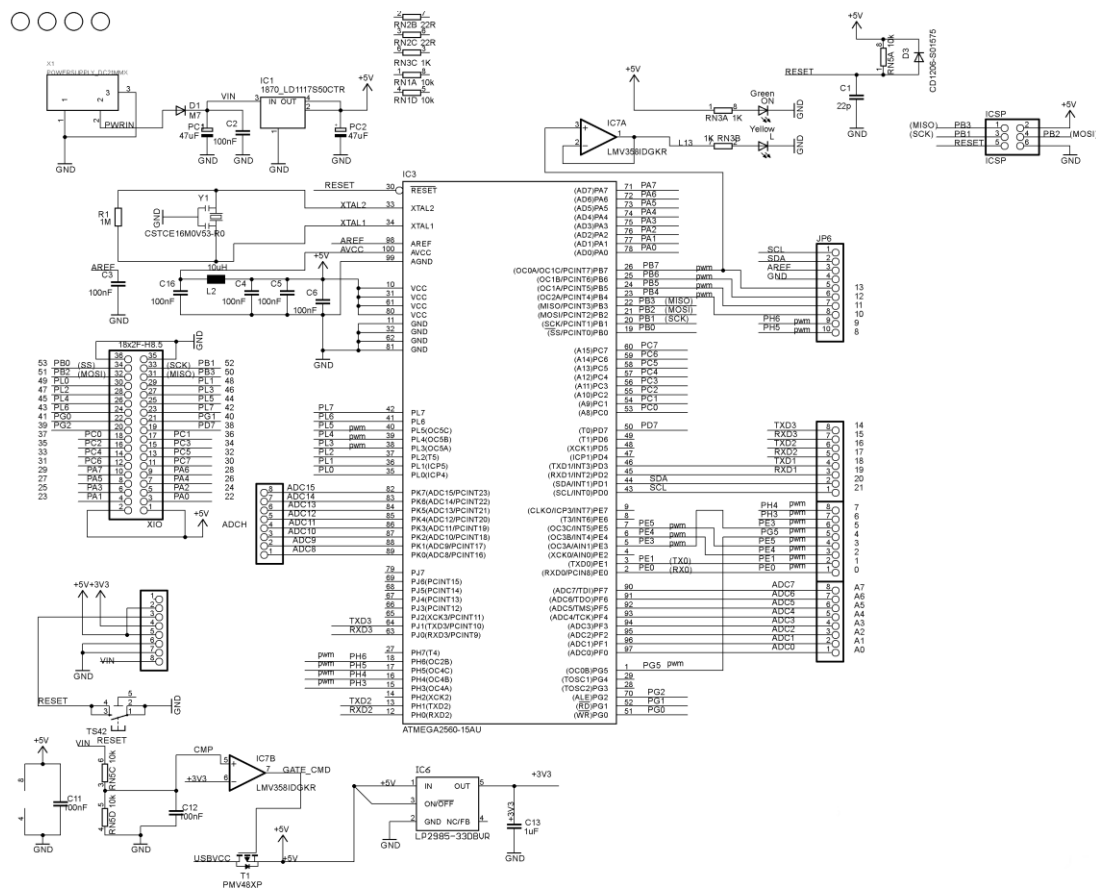


Рисунок 2.7 – Принципова схема основної процесорної частини МК

На схемі відображені елементи:

- мікроконтролер ATmega 2560-15 AU ;
- кварцовий резонатор на 16МГц, що задає тактову основну частоту;
- інтегральний стабілізатор на 5В типу LD 1117 S 50 або аналог;
- інтегральний стабілізатор на 3,3В типу LP 2985-33 або аналог [11];
- індикатори на світлодіодах: зелений показує наявність живлення МК, жовтий світлодіод управляється від порту A13 через повторювач на LM358, другий повторювач цієї мікросхеми задіяний як компаратор, що виявляє наявність живлення на VIN і дає команду ключовому транзистору Т 1 на подачу живлення 5 завдяки захисному зворотному діоду цього транзистора можливе живлення плати від USB;
- елементи обв'язки мікроконтролера, що забезпечують нормальний режим роботи мікропроцесора.

Принципова схема додаткової частини плати Arduino MEGA 2560, що виконує роботу з USB , зображена на рис. 2.8.

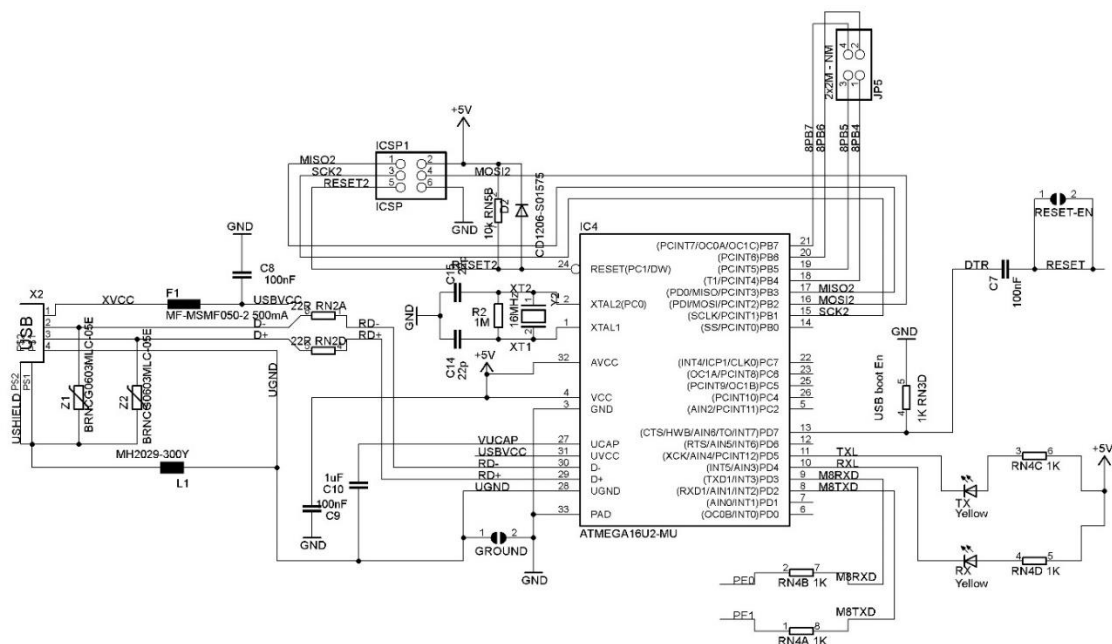


Рисунок 2.8 – Принципова схема контролера USB МК

На схемі зображені елементи: мікроконтролер ATmega 16 U 2- MU ; кварцовий резонатор на 16МГц, що задає тактову основну частоту цього

контролера; USB -порт з елементами обв'язування та захисту від перенапруги; світлодіоди, що показують режими роботи мікроконтролера та кілька елементів обв'язування мікроконтролера, що забезпечують нормальний режим роботи мікропроцесора.

На рис. 2.9 наведено принципову схему плати управління та індикації Keypad LCD Shield [7]. На схемі зображені:

- роз'єми плати розширення J1- J7 та ICSP ;
- світлодіод індикації живлення VCC ;
- кнопка RST , підключена до виходу RST плати Arduino , необхідна для скидання мікроконтролера, так як плата розширення закриває кнопку RESET на платі Arduino ;
- кнопки RIGHT , UP , DOWN , LEFT та SELECT з резисторною матрицею підключаються до аналогового входу AD 0 плати Arduino;
- елементи індикатора LCD 1602: резистор регулювання контрастності зображення RP 1, транзистор комутації підсвічування Q1 виводу D 10 плати Arduino .

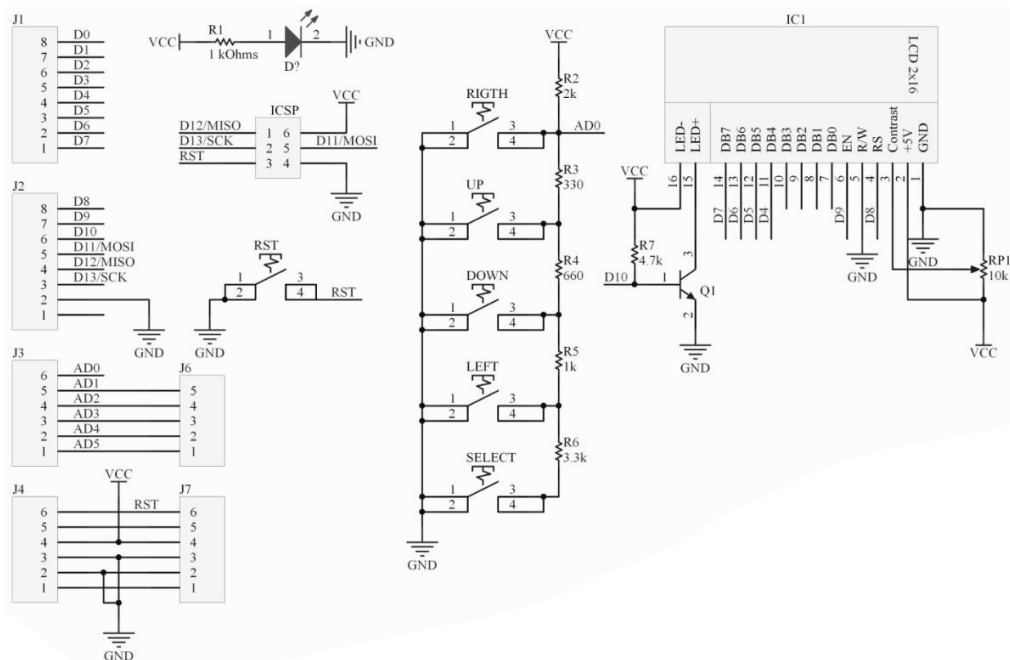


Рисунок 2.9 – Принципова схема плати індикації та управління

На рис. 2.10 зображено частину схеми керування силовим навантаженням.

Схема виконана на 4 симмісторах Q1 типу BTA 16-600V з оптичною розв'язкою D3 на диністорних оптопарах MOC3021S та індикацією стану входу на світлодіоді D4 [13].

Резистори R4-R6 служать для обмеження струму через напівпровідникові елементи і задання робочих режимів. На оптопарі D2, діодному мості D1 і обмежувальних резисторах R1-R3 побудована схема детектора нуля, необхідна для правильного перемикання симмістора під час переходу мережевої напруги живлення навантаження через нуль.

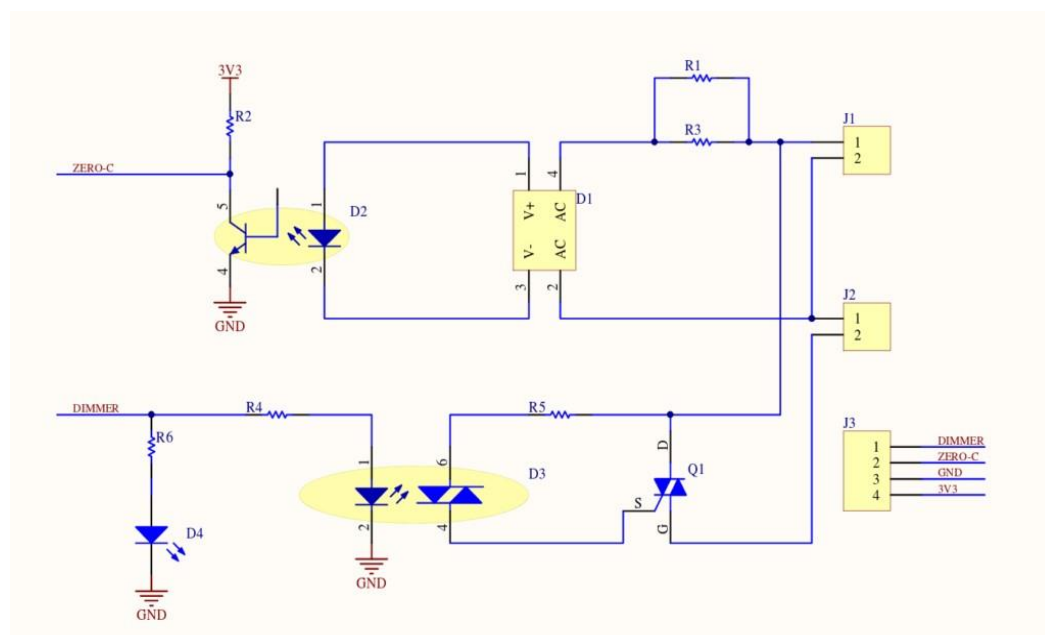


Рисунок 2.10 – Частина принципової схеми плати комутації навантажень

На рис. 2.11 наведено принципову схему імпульсного блоку живлення Coreset SM-PLG06A-09 [3], зібраного за класичною трансформаторною схемою одноктного ПППН на спеціалізованій інтегральній мікросхемі з вхідним фільтром ЕМІ, що зменшує проникнення високочастотних пульсацій

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.115.00.00 ПЗ

в мережу живлення, шляхом регулюванням вихідної напруги через оптопару і інтегральний стабілізатор типу TL .

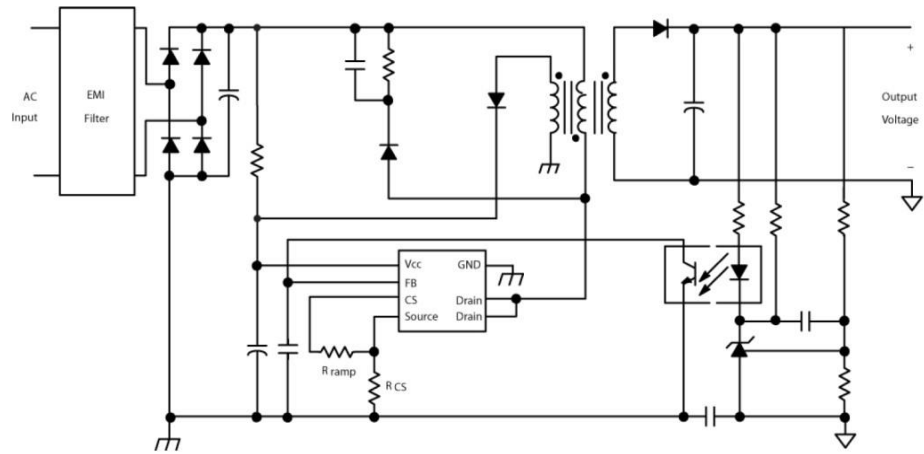


Рисунок 2.11 – Принципова схема імпульсного блоку живлення

На рис. 2.12 наведено призначення контактів плати комутації навантажень

- Power
- Control
- GND
- PWM

- VCC
- GND
- ZERO
- DMR1
- DMR2
- DMR3
- DMR4

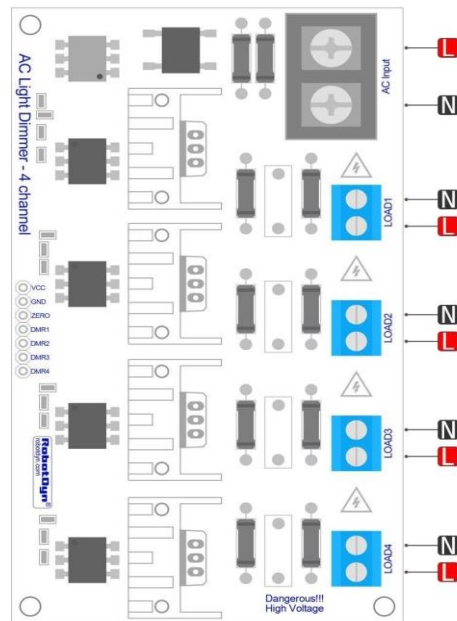


Рисунок 2.12 – Призначення контактів плати комутації навантажень

Маючи необхідні частини пристрою можна сформувавши принципову схему системи управління силовими розетками. Для тестового варіанту зупинемось поки на управлінні чотирма розетками. При необхідності кількість розеток можна збільшити. Схема наведена на листі КС КРБ 123.115.00.00 ЕЗ.

### 2.3 Обґрунтування вибору програмного забезпечення

Для функціонування системи керування розетками необхідно написати програму, яка виконує такі функції:

- вибір каналу керування;
- встановлення часу вмикання таймера в межах доби;
- встановлення часу вимкнення таймера в межах доби;
- встановлення періодичності повторення таймера;
- збереження налаштувань таймера в енергонезалежну пам'ять;
- перехід з режиму налаштування в режим індикації після закінчення таймера індикації автоматично або за допомогою команди з панелі керування вручну;
- індикація стану всіх таймерів на одному екрані.

Для МК Arduino пишеться програма (sketch, прошивка) мовою програмування C/C++ серед програмування Arduino IDE [6], що складається з:

- програмний модуль (функція) формування хвилинних імпульсів, в якому задіюється апаратний 16 бітний таймер Atmega 2560, через функцію millis (), який відраховує мілісекунди і, після досягнення 60 секунд, записує відповідне значення в змінну хвилин, перезапускає малий хвилину;
- програмний модуль (функція) формування добових імпульсів, який відраховує хвилини з даних першого таймера і після досягнення 60 хвилин записує відповідне значення змінну годин, після досягнення значення 1440 реєструється добовий інтервал, а після досягнення значення

встановленого в змінну повторення добового циклу;

- програмний модуль обчислення поточного часу із змінних годин та хвилин з урахуванням початкової установки годинника;
- 4 програмні модулі (функції) лічильників електронних таймерів з формуванням сигналів управління блоком комутації навантажень, порівнюють поточне значення змінних хвилин і діб із заданим і дають команди на включення та вимкнення навантажень;
- програмний модуль (функція) формування інформації на індикаторі МК, що обробляє формування відображення символів у навігації по меню налаштування пристрою;
- програмний модуль (функція) кнопок панелі програмування МК, обробляє аналоговий сигнал із панелі кнопок та забезпечує керування навігацією по меню налаштування пристрою;
- програмний модуль (функція) запису/відтворення значень змінних в незалежну пам'ять.

Індикатор показує час у годинах та хвилинах, що розділяє їх знак «:» блимає з періодичністю в 1 секунду, що говорить про основний режим роботи індикатора. У режимі налаштування часу знак «:» не блимає, але блимає цифра або знак, які змінюємо.

Також було необхідно реалізувати керування з допомогою веб-інтерфейсу.

Для цього було використано програмне забезпечення описане нижче.

Yii 2 фреймворк - це високоефективний заснований на компонентній структурі PHP -фреймворк для розробки масштабних web - додатків.

Даний фреймворк має високу швидкодію. Реалізований за моделлю Model - view - controller ( MVC ). Має набори класів Data Access Object та ActiveRecord для роботи з базами даних.

Також можна використовувати різні бібліотеки для розширення функціональних можливостей.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32



Єдиний значний недолік у використанні даного фреймворку полягає в тому що весь код, який включає фреймворк, створює навантаження на сервер.

В якості бази даних для зберігання параметрів при розробці програмного забезпечення було використано MySQL .

Структура бази даних наведена на рис. 2.13

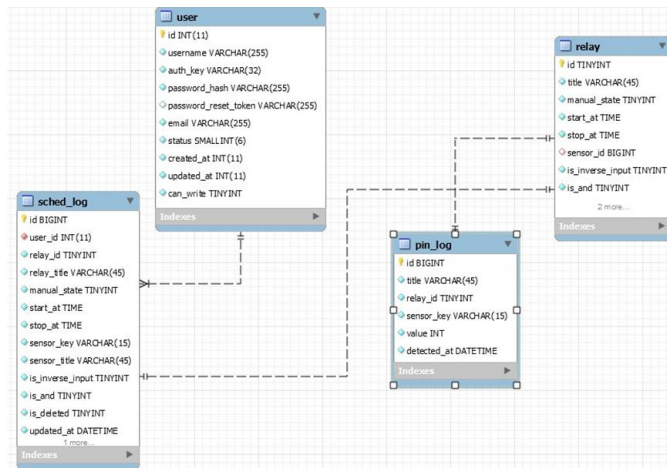


Рисунок 2.13 - Структура бази даних

У таблицю user заносяться користувачі, які зареєстровані і мають доступ до до системи керування розетками.

Таблиця Sched\_log використовується для запису журналу керування силовими розетками.

У таблицю relay записується журнал розкладу ввімкнення і вимкнення силових розеток.

У таблицю Pin\_log записуються зміни стану силових розеток.

Відповідно для керування розетками необхідно передбачити наступні вікна:

- розклад увімкнення силових розеток;
- журнал розкладу;
- сторінка адміністрування.

## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Реалізація та моделювання проектних рішень

#### 3.1.1 Реалізація апаратної складової комп'ютеризованої системи

Для пристрою вибрано вертикальний корпус розмірами 150x200x50мм. Передня панель виконана з вирізами під розетки, загальний вимикач живлення, індикатор та підпружинені кнопки. Застосування передньої металевої панелі з кнопок управління зменшує можливість статичного пробоя МК при програмуванні таймерів. Частина передньої панелі відкривається для надання доступу до USB порту МК для програмування із зовнішнього персонального комп'ютера.

Для підключення навантажень застосуємо 4 приховані панельні розетки промислового типу з кришками (рис.3.1). Розмір розеток 50x50мм з урахуванням шарніру кришки 65мм, глибина установки 36мм, висота кришки 15мм. Діаметр отвору панелі під установку розетки 43мм. Розетки кріпляться до панелі чотирма болтами М4, через 4 отвори діаметром 5мм, шарнірами кришок до індикатора МК. Передбачено встановлення чотирьох керованих розеток.



Рисунок 3.1 – Набір розеток 220В/16А змонтованих в єдиному корпусі

Розміри основних плат, з відмітками отворів кріплення діаметром 3,2 мм

					<b>КС КРБ 123.115.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Головатий А.			<i>Практична частина</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		Жаровський					34	
<i>Реценз.</i>						<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		Тиш Є.В.						
<i>Затверд.</i>		Осухівська Г.М.						

під гвинт М3, наведені на рисунках 3.2-3.4.

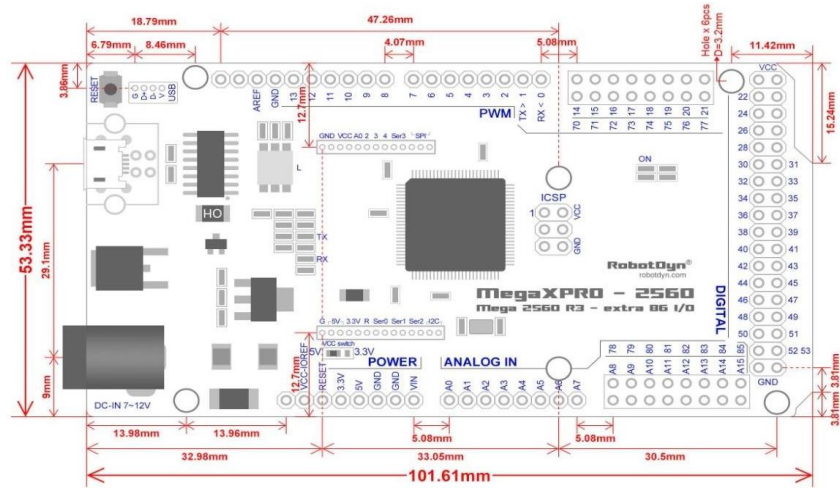


Рисунок 3.2 – Модуль мікроконтролера

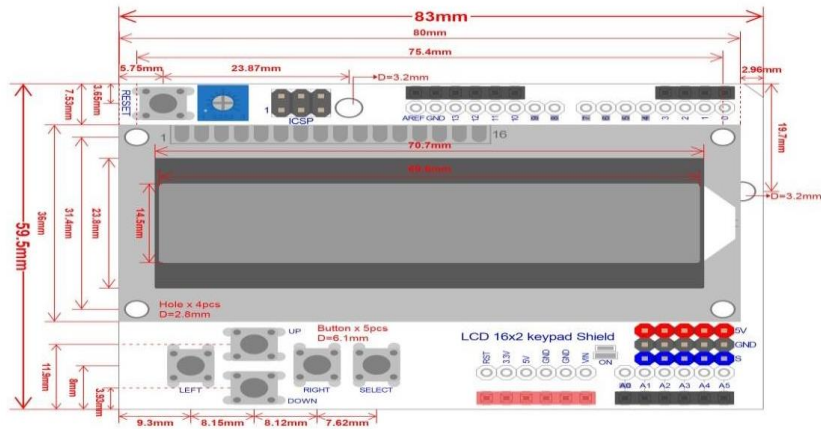


Рисунок 3.3– Модуль індикатора та панелі керування

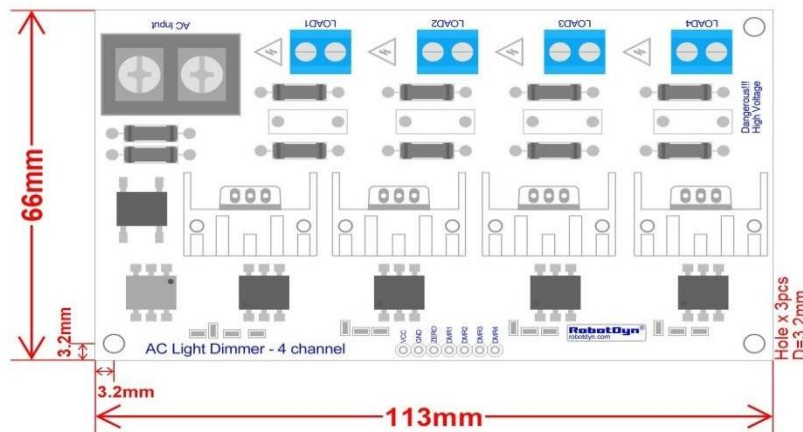


Рисунок 3.4 – Модуль комутатора навантажень

Усі плати всередині корпусу розташовуються вертикально на

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

монтажних стійках, причому, плати індикатора та МК кріпляться всередині корпусу до передньої панелі корпусу. Плата комутації навантажень та блок живлення кріпляться на передній кришці корпусу. Плата індикації встановлюється в роз'єм плати МК (рис.3.5) як плата розширення ( Shield ). Плата індикації виводиться на верхню частину корпусу. Для цього передбачено необхідний виріз 60x83 мм.

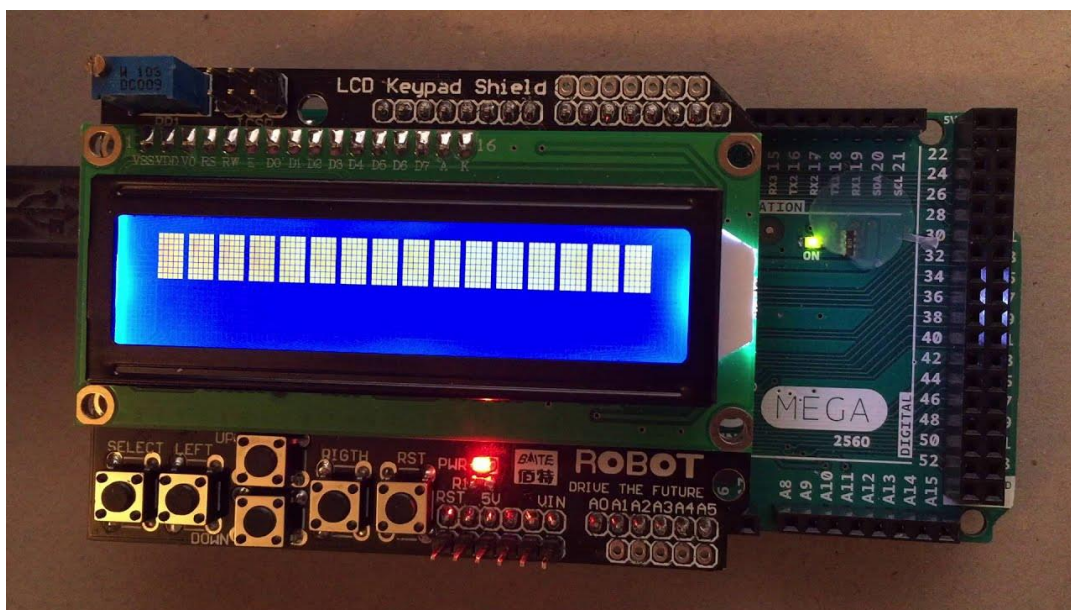


Рисунок 3.5 – Вид плати індикації підключеної до плати МК

З'єднання МК з платою комутації навантажень здійснюється плоским 7-провідним кабелем з наконечниками DuPont «тато-мама». Живлення 9В від блока живлення підключається до мікроконтроллера штекером DC 5.5\*2,1.

Кабель живлення системи керування силовими розетками підключається через 3-х контактний роз'єм типу IEC 320- С 14 (рис.3.6) з вимикачем та колодкою запобіжника моделі Multicomp JR-101-1-FRSG-03. На бічній панелі під нього робиться прямокутний отвір розмірами 28мм на 48мм.



### Specifications

Rating	: 10A 250V AC
Withstand Voltage	: 2,000V AC 1 minute
Insulation Resistance	: More than 100MΩ (at 500V DC)
Max. Operating Temperature	: -25°C to +85°C
IEC Connector Type	: C14

### Diagram

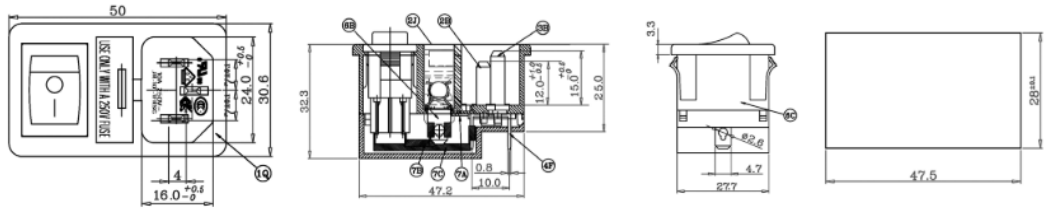


Рисунок 3.6 – Модуль для підключення кабелю живлення

### 3.1.2 Реалізація програмної складової системи

Підключаємо бібліотеки дисплея, енергонезалежної пам'яті, математичних формул і мережі (рис. 3.7).

```
#include < LiquidCrystal . h >
#include <EEPROM.h>
#include <MATH.h>
#include <EtherCard.h>
```

Рисунок 3.7 – Лістинг підключення бібліотек

Далі необхідно активувати і призначити виходи мікроконтролера і задати необхідні змінні та зарезервувати під них пам'ять (рис.3.8).

```
LiquidCrystal lcd (12, 11, 9, 8, 7, 6, 5, 4, 3, 2);
int rtcM=0, rtcH=0, rtcS=0; //Початкові значення змінних
часу
int p1set=0, p2set=0, p3set=0, p4set=0, t1set=0, t2set=0,
t3set=0, t4set=0; //Початкові значення установок таймерів
int timer1=0, timer2=0, timer3=0, timer4=0, period1=0,
period2=0, period3=0, period4=0; //лічильники таймерів
int D=0, Dig=1, Sel=0, Dx=0; //Початковий екран Display(0...4),
перша цифра
bool b=0; //змінна відображення інформації на екрані
bool t1=0, t2=0, t3=0, t4=0; //Змінні стани таймерів
```

Рисунок 3.8 - Призначення виходів і оголошення змінних

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.115.00.00 ПЗ

Арк.

37



Також наведу лістинг основних налаштувань мікроконтролера (рис.3.9).

```
void setup ()
{
  lcd . begin (16, 2); // ініціалізуємо дисплей: 2 рядки по 16
  символів
  readSet (); // зчитуємо раніше збережені налаштування
  pinMode (22, OUTPUT ); // Призначення порту 22 на вихід
  управління навантаженням таймера 1
  pinMode (24, OUTPUT ); // Призначення порту 24 на вихід
  управління навантаженням таймера 2
  pinMode (26, OUTPUT ); // Призначення порту 26 на вихід
  управління навантаженням таймера 3
  pinMode (28, OUTPUT ); // Призначення порту 28 на вихід
  управління навантаженням таймера 4
}
```

Рисунок 3.9 - Призначення виходів і оголошення змінних

Після чого необхідно створити ряд функцій які будуть відповідати за натискання кнопок void xKey (), обробку лічильників часу void timersWork (), керування перемиканням екранів void setPlay () .

Екрани формуємо через функції екранів. Усього буде 4 основних типи екранів і три екрани налаштування часу та таймерів. Перший основний екран відображає час у годинах та хвилинах усіх чотирьох таймерів із зазначенням маркеру роботи таймера.

Керування навантаженнями здійснюється через плату розширення, на яку приходить цифровий сигнал про включення або вимкнення відповідного виходу. Формуємо функцію управління станом виходів мікроконтролера та встановлюємо у відповідний стан булеві змінні таймерів (рис. 3.10).

```

void setOut () { // встановлення рівня виходу
  if ( timer 1>0) { digitalWrite (22, HIGH ); t 1=1;}
  //якщо час таймера 1 більше 0 (включено) включити вихід по 22 порту
  else { digitalWrite (22, LOW ); t 1=0;}
  //якщо час таймера 1 дорівнює 0 (вимкнений) вимкнути вихід по 22
  порту
  if ( timer 2>0) { digitalWrite (24, HIGH ); t 2=1;}
  //якщо час таймера 2 більше 0 (включено) включити вихід по 24 порту
  else { digitalWrite (24, LOW ); t 2=0;}
  //якщо час таймера 2 дорівнює 0 (вимкнений) вимкнути вихід по 24
  порту
  if ( timer 3>0) { digitalWrite (26, HIGH ); t 3=1;}
  //якщо час таймера 3 більше 0 (включено) включити вихід по 26 порту
  else { digitalWrite (26, LOW ); t 3=0;}
  //якщо час таймера 3 дорівнює 0 (вимкнено) вимкнути вихід по 26 порту
  if ( timer 4>0) { digitalWrite (28, HIGH ); t 4=1;}
  //якщо час таймера 4 більше 0 (включено) включити вихід по 28 порту
  else { digitalWrite (28, LOW ); t 4=0;}
  //якщо час таймера 4 дорівнює 0 (вимкнено) вимкнути вихід по 28 порту
}

```

Рисунок 3.10 - Функція управління станом виходів мікроконтроллера

Для запобігання скидання налаштувань при випадковому зникненні живлення застосовані функції збереження основних налаштувань таймерів та годинників у енергонезалежну пам'ять void saveSet (). Але найкращим рішенням, звичайно ж, буде забезпечення безперебійного живлення плати мікроконтроллера із підключенням акумулятора напругою 7,4В з ланцюгом підзарядки від блоку живлення.

Повний текст програми наведено в додатку Б.

## 3.2 Тестування системи

Перед тестуванням системи було проведено моделювання її роботи. Для цього було зібрано її макет по наступній схемі (рис.3.11).

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

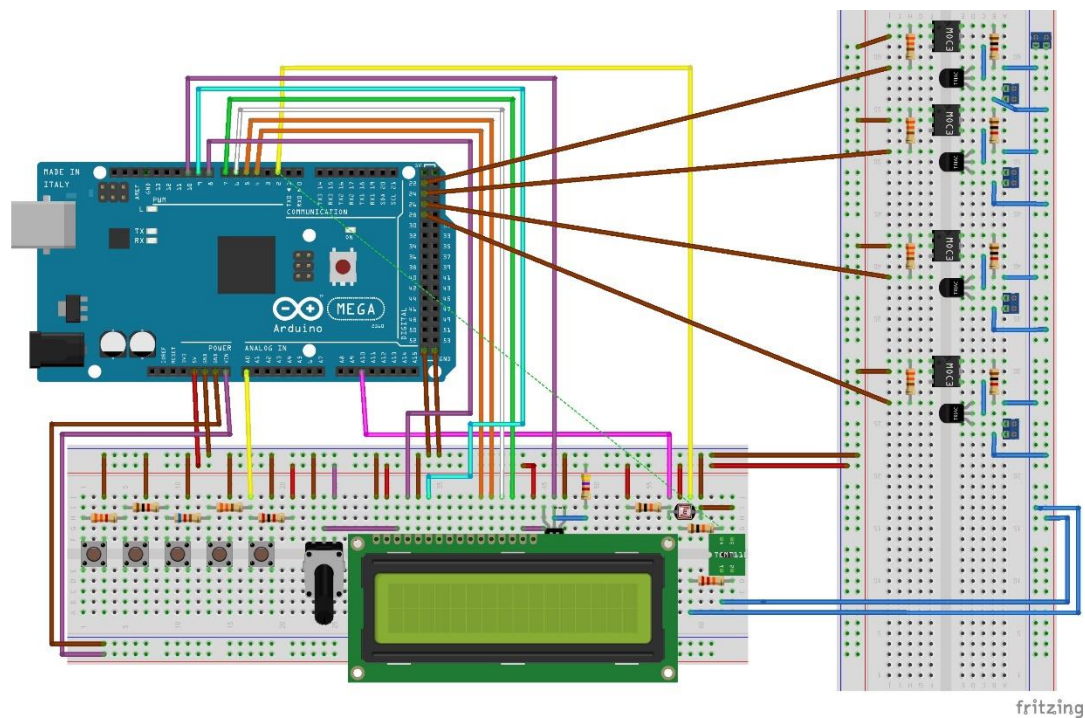


Рисунок 3.11 – Макет для моделювання роботи пристрою

Складена програма відкомпільована в Arduino IDE ver.2.0 , тобто не містить синтаксичних помилок (рис.3.12).

```

Output
Sketch uses 9430 bytes (3%) of program storage space. Maximum is 253952 bytes.
Global variables use 218 bytes (2%) of dynamic memory, leaving 7974 bytes for local variables. Maximum is 8192 bytes.

-----
Compilation complete.
  
```

Рисунок 3.12 – Звіт про компіляцію програми

Програма завантажена в симулятор та перевірена у роботі. Усі функції системи керування працюють відповідно до завдання. Нижче наведено приклади скрінів роботи програми.

В основному режимі роботи індикатора можливі кілька видів екрану:

1) Основний екран - індикатор пристрою показує час усіх чотирьох таймерів ( T1- T4) у годинах та хвиликах « hh : mm » та поточний стан кожного таймера (ввімкнено « \* » або вимкнено « ») (рис.3.13).



Натисканням кнопок LEFT/RIGHT можна перемикається між видами екрана стану таймерів пристрою по колу. Натисканням кнопок UP / DOWN можна перейти до екрана поточного часу.

Натисканням кнопки SELECT можна перейти до налаштувань таймерів по 4 каналах пристрою;

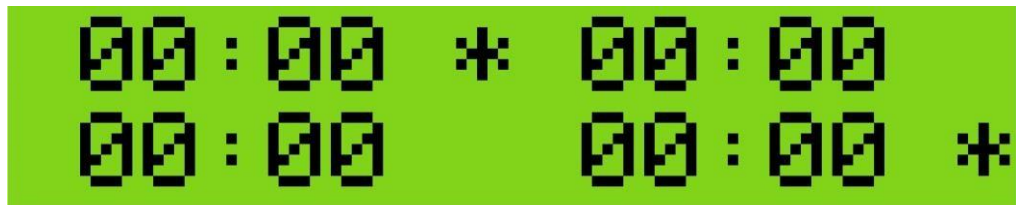


Рисунок 3.13 – Основний екран

2) Екран поточного часу – індикатор показує поточний час « Now time » у годинах та хвилинах « hh : mm » і час, що минув з моменту включення пристрою в днях " uptime \*\* day " (рис. 3.14).

Натисканням кнопки SELECT можна перейти до налаштувань поточного часу, де натисканням кнопок LEFT / RIGHT можна перемикається між цифрами, а натисканням кнопок UP / DOWN можна їх змінювати. Відображення цифри, що змінюється в даний момент часу, буде блимати з періодичністю в 1 секунду, знак «:» не блимає. Натисканням кнопки SELECT встановлення поточного часу записується в незалежну пам'ять;



Рисунок 3.14 – Екран поточного часу

3) Індикатор пристрою показує стан (ввімкнений « ON » або вимкнений « OFF ») одного поточного таймера « T \*» у хвиликах «\*\*\*\* m », та періодичність включення таймера « Period \*\* day » (рис.3.15) або час після останньої зміни стану « PASTtime \*\*\*\* m » на рис. 3.16 або час, що залишився до наступної зміни стану таймера « END time \*\*\*\* m » на рис.3.17;

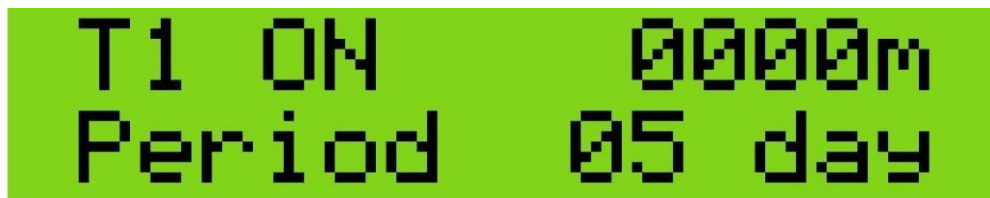


Рисунок 3.15 – Екран тривалості та періодичності включення поточного таймера



Рисунок 3.16 – Екран часу з моменту включення поточного таймера



Рисунок 3.17 – Екран часу поточного таймеру, що залишився.

Натисканням кнопки SELECT можна перейти до налаштувань таймерів пристрою (рис.3.18), де натисканням кнопок LEFT / RIGHT можна перемикатися між таймерами, а натисканням кнопок UP / DOWN можна змінювати їх налаштування. Відображення числа, що змінюється в даний момент часу, буде блимати з періодичністю в 1 секунду. Повторне натискання кнопки SELECT перемикає екран налаштування таймера між тривалістю та

періодичністю увімкнення таймера. Натисканням кнопки SELECT налаштування таймера записуються в незалежну пам'ять і пристрій переходить до відображення основного екрана (рис.3.13);

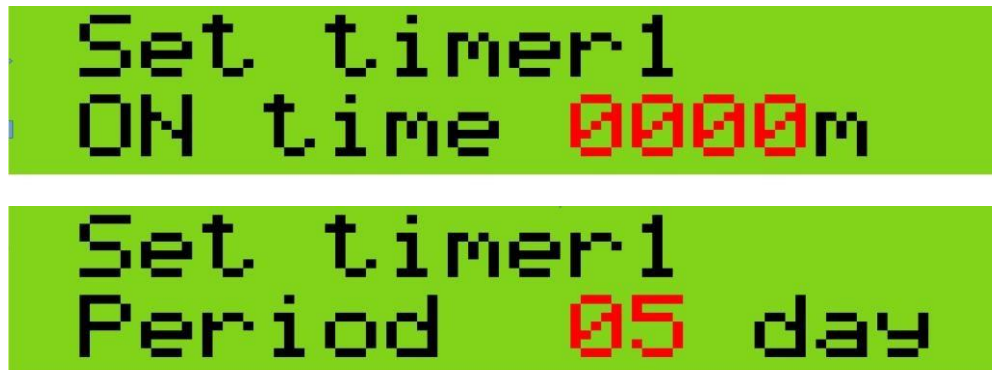


Рисунок 3.18 – Екрани налаштування таймерів

На рис. 3.19 відображено алгоритм роботи екранного меню у спрощеному вигляді – відображення стану та налаштувань 2 та 3 таймерів аналогічно 1 та 4 таймерів, тому для наочності вони зображені сірим прямокутником.

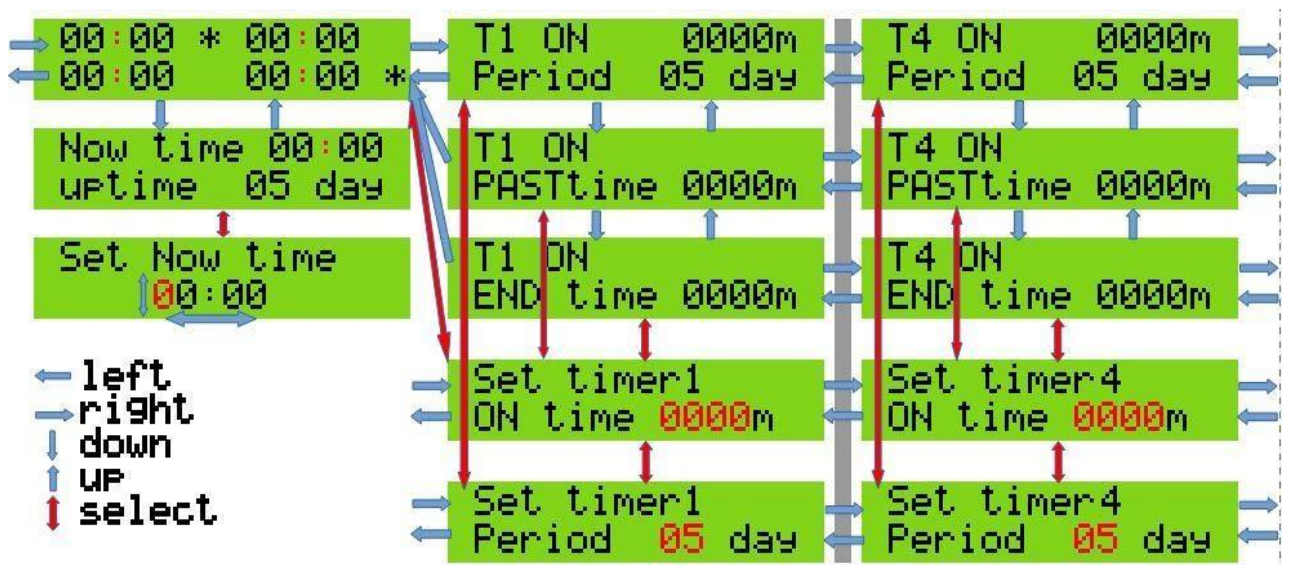


Рисунок 3.19 – Алгоритм роботи екранного меню

Синіми стрілками зображені переходи між екранами при натисканні кнопок LEFT / RIGHT та UP / DOWN . Червоними стрілками зображено

перехід між екранами під час натискання кнопки SELECT. Червоним шрифтом на екранах зображені області, що блимають із частотою 1 Гц.

Також були розроблений веб-інтерфейс для керування з віддаленого ПК. Результати наведені на листі КС КРБ 123.115.00.00.

					<i>КС КРБ 123.115.00.00 ПЗ</i>	Арк.
						44
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Ергономічні проблеми безпеки життєдіяльності

Основним завданням ергономіки є забезпечення ефективної взаємодії людини і техніки, щоб перейти від техніки безпеки до безпечної техніки, яку ми використовуємо як у виробничій, так і побутовій сферах. Це один з основних напрямків ергономіки.

Основним об'єктом ергономіки є «людино-машинні» системи. Тому основне завдання ергономіки – вивчення зв'язків між елементами «людино-машинних» систем, розробка методів сумісності основного її компонента – людини з іншими середовищами та машинами, технікою [14].

Завдання ергономіки:

- вивчення закономірностей взаємодії людини з технічними системами та навколишнім середовищем;
- формування принципів побудови системи «людино-машинних» систем та алгоритмів дії у них людини-оператора;
- розробка основ проектування діяльності людини-оператора з врахуванням специфіки експлуатації технічних систем та факторів навколишнього середовища;
- розробка перспективних форм праці людини і пов'язаних з нею технічних систем, факторів навколишнього середовища;
- розробка методів дослідження, проектування та експлуатації «людино-машинних» систем, які забезпечують безпеку людини, ефективність праці.

Проблемами взаємодії людини та машини займається також інженерна психологія, яка вивчає закономірності процесів інформаційної взаємодії людини у системі «людино-машинних» систем.

					<b>КС КРБ 123.115.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Головатий А.			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Перевірів</i>		Жаровський					45	
<i>Консульт.</i>		Пилипець М.І.				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		Тиш Є.В.						
<i>Затверд.</i>		Осухівська Г.М.						

У системі «людино-машинних» систем завжди є 3 елементи:

- предмет праці;
- засоби праці;
- суб'єкт праці.

Найменшою цільною одиницею, де наявні вказані елементи, є місце праці. Місце праці – це зона, де відбувається трудова діяльність людини. Місце праці обладнане засобами відображення інформації, органами керування та допоміжним обладнанням.

Організацією місця праці називається проведення системи заходів щодо його обладнання засобами та предметами праці і їх розташуванням у визначеному порядку з метою досягнення [15]:

- оптимізації умов трудової діяльності;
- безпеки праці;
- максимальної ефективності;
- комфортності роботи людини.

До робочого місця ставляться такі вимоги:

- достатній робочий простір, який дає змогу працюючій людині здійснювати необхідні рухи та переміщення;
- достатні фізичні, зорові та слухові зв'язки між людиною та обладнанням, а також між людьми під час виконання спільного трудового завдання;
- необхідний рівень освітлення;
- наявність необхідних засобів захисту;
- оптимальне розташування робочих місць, а також безпечні та достатні проходи для працюючих людей.

При організації робочого місця враховують основні антропометричні дані людини. Найважливішою характеристикою робочого місця є зона досягнення моторного поля. Моторне поле – це простір робочого місця, в якому розміщені органи керування та інші технічні засоби, в якому людина здійснює рухові дії для виконання робочого завдання.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Ергономіка виробила конкретні вимоги до антропометричних показників обладнання [15].

Характеристика пульта:

- загальна висота: «сидячи»– 1650 мм, «стоячи» – не більше ніж 1300 мм;
- висота розміщення органів керування для положення «сидячи» 530 - 1040 мм, стоячи - 1000 - 1500 мм.

Характеристики крісла:

- форма сидіння-квадратна;
- форма спинки - прямокутна вгнута;
- розмір сидіння - 400x400 мм, спинки - 300x120 мм;
- кут нахилу сидіння назад - 50 - 60°;
- кут нахилу спинки - 50 - 100°;

Розміри вільного місця для ніг:

- висота - не менше 600 мм;
- ширина - не менше 500 мм;
- глибина - не менше 400 мм.

Досягнення органів керування по горизонталі – півколо радіусом 600 мм. Встановлені також відстань між органами керування, їх розміри, зусилля переміщення, величина переміщення, напрямок переміщення.

Для операторів, які працюють з екранами дисплеїв та інших індикаторів, можуть бути рекомендовані такі режими праці та відпочинку. Тривалість безперервної праці не повинна перевищувати 4-6 год. В іншому випадку працездатність через втому зору раптово знижується.

Оператор, який стежить за екраном індикатора, найуважніше працює протягом перших 30 хв чергування. А далі, внаслідок втоми зорового аналізатора, кількість помилок зростає майже в два рази та залишається незмінною до кінця другої години. Потім спостерігається нове зростання кількості помилок через загальну втому оператора. Тому для підтримки високої ефективності праці може бути рекомендований 30-хвилинний період чергування з наступною 30-хвилинною перервою.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		



#### 4.2 Вимоги пожежної безпеки при гасінні електроустановок

В кваліфікаційній роботі розроблено комп'ютеризовану систему керуванням силовими розетками. Тому питання пожежної безпеки організацій у яких використовуються електроустановки є дуже важливим. Пожежна безпека забезпечується шляхом здійснення організаційно-технічних та інших заходів з попередження виникнення пожеж, зменшення можливих матеріальних збитків, забезпечення безпеки людей, зниження негативних екологічних наслідків, створення умов для успішного гасіння пожеж та швидкого виклику пожежних підрозділів, а також евакуації з території виникнення та ймовірного розповсюдження пожежі людей, матеріальних цінностей і документів.

При виникненні пожежі можна виділити два методи, які застосовуються для гасіння електроустановок:

- гасіння електроустановок відключених від напруги мережі;
- гасіння електроустановок, які знаходяться під напругою.

У разі виникнення пожежі на електроустановці особа, яка першою виявила факт загорання, повинна негайно повідомити відповідальних за пожежну безпеку осіб та керівника для уникнення подальшого загорання. Гасіння електрообладнання під напругою із застосуванням ручних стволів повинне виконуватися за умови [16]:

- застосування ефективних прийомів і способів подачі в зону горіння вогнегасних речовин;
- дотримання електробезпечних відстаней від електрообладнання, яке знаходиться під напругою, до пожежників, які використовують ручні пожежні стволи;
- забезпечення надійного заземлення пожежних автомобілів і стволів;
- використання індивідуальних ізолюючих електрозахисних засобів під час гасіння пожежі електроустановок без зняття напруги.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення безпеки пожежників та персоналу, який бере участь у гасінні пожежі електроустановок під напругою, застосовуються ізолюючі індивідуальні електрозахисні засоби (діелектричні килими, калоші, діелектричні рукавиці). Кількість індивідуальних ізолюючих захисних засобів та заземлень і місця їх зберігання затверджуються керівниками енергетичних об'єктів враховуючи подачу вогнегасних засобів на електроустановки, які перебувають під напругою [17].

При гасінні пожежі на електроустановках під напругою потрібно застосовувати прийоми та засоби подачі в зону горіння вогнегасних речовин, що забезпечують ефективне гасіння пожежі і безпечну роботу пожежників.

В якості вогнегасних речовин під час гасіння електроустановок, які знаходяться під напругою, доцільно застосовувати: розпилені порошкові суміші й інертні гази, струмені води, комбіновану суміш, яка являє собою розпилену воду з порошком. Застосування усіх видів піни під час гасіння електроустановок під напругою за участю людей ручними засобами забороняється, через те що піна й розчин піноутворювача мають підвищену електропровідність в порівнянні з розпиленою водою.

При гасінні пожежі на електроустановках, які перебувають під напругою до 220 кВ включно, тривалість перебування пожежників на позиціях не лімітується. Заземлення ручних пожежних насосів і стволів пожежних автомобілів при гасінні пожеж на електроустановках, які знаходяться під напругою, повинно виконуватись за допомогою гнучких мідних провідників з поперечним перерізом не менше 12 мм<sup>2</sup>, оснащених спеціальними струбцинами для під'єднання до заземлених конструкцій: шурфів, обсадних труб артезіанських свердловин, металевих опор повітряних ліній електропередач, гідрантів водогінних мереж [16].

Під час виникнення пожежі в приміщенні де буде експлуатуватись комп'ютеризована система управління силовими розетками, яка розроблена в кваліфікаційній роботі, необхідно дотримуватись вимог пожежної безпеки при гасінні електроустановок.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Кваліфікаційна робота виконана відповідно до завдання на розробку комп'ютеризованої системи керування силовими розетками.

Особливістю даної роботи є складання багатофункціонального програмованого пристрою на платформі швидкого прототипування Arduino з програмуванням унікального алгоритму роботи пристрою мовою програмування C/C++ в інтегрованому середовищі розробки Arduino IDE з подальшою симуляцією роботи пристрою.

Поставлені завдання були виконані у повному обсязі:

- визначено основні функції систем керування силовими розетками та знайдено функціональні аналоги;
- розроблено чотириканальну комп'ютеризовану систему керування розетками відповідно до завдання на проектування.

Розроблений пристрій може бути використаний для малої автоматизації на виробництві в малому та середньому бізнесі, наприклад, для керування електричними приладами теплиці (освітлення, вентиляція, обігрів та полив рослин).

Шляхом зміни програми пристрою можна створити різну функціональність засобів промислової автоматизації, у тому числі керувати пристроями розумного будинку.

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Evans, BW Arduino Programming Notebook. - Creative Commons: San Francisco, 2007. 40 p.
2. Різні типи плат Arduino та їх використання URL: <https://uk.fmuser.net/content/?12409.html> (дата звернення: 15.05.2024).
3. SM-PLG06A Specifications URL: <http://www.kosmodrom.com.ua/pdf/PLG06A-SERIES.pdf> (дата звернення: 15.05.2024).
4. Margolis, M. Arduino Cookbook, Second Edition. O'Reilly Media, Inc., 2012. 724 p.
5. Романов Д.В., Осухівська Г.М., Паламар А.М. Система управління зовнішнім освітленням на основі Інтернету речей. Актуальні задачі сучасних технологій : збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24-25 листопада 2021 року), Тернопіль: ТНТУ, 2021. С. 120.
6. Schmidt, M. Arduino: A Quick-Start Guide / M. Schmidt - The Pragmatic Bookshelf - Dallas, USA: 2011. 284 p.
7. Романов Д.В., Осухівська Г.М., Паламар А.М. Функціональна схема системи керування зовнішнім освітленням на основі технології LoRa. Матеріали ІХ науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 8–9 грудня 2021 року), Тернопіль: ТНТУ, 2021. С. 124.
8. Schwartz M. Internet of Things with the Arduino Yún Packt Publishing, 2014. 112 p.
9. Wiley, J. Exploring Arduino: Tools and Techniques for Engineering Wizardry/J.Wiley, - Indianapolis, Canada, 2020. 491p.
10. Слюз І. Принципи та основні етапи комплексного тестування комп'ютерної інформаційної системи / І. Слюз, Р. Жаровський // Матеріали X

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

науково-технічної конференції „Інформаційні моделі, системи та технології“, 7–8 грудня 2022 року. — Т. : ТНТУ, 2022. — С. 94

11. Ясінський Р.В., Осухівська Г.М., Паламар А.М. Апаратно-програмна система для регулювання мікроклімату теплиць. Матеріали X науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 7-8 грудня 2022 року), Тернопіль: ТНТУ, 2022. С. 102.

12. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU.Tern.: TNTU. 2024. Vol 109. No 1. P. 54–65.

13. Yatsyshyn V., Pastukh O., Zharovskyi R., Shabliy N. Software tool for productivity metrics measure of relational database management system. Mathematical Modeling. No 1 (48). 2024. p. 7-17.

14. Ергономічні принципи безпеки URL: [https://pidru4niki.com/12281128/bzhd/ergonomichni\\_printsipi\\_bezpeki](https://pidru4niki.com/12281128/bzhd/ergonomichni_printsipi_bezpeki) (дата звернення: 13.06.2024).

15. Бедрій Я. І. Безпека життєдіяльності: навч. Посібник. Київ.: Кондор, 2009. 286 с.

16. Желібо Є. П. Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності. Навчальний посібник. К.: Каравела, 2004. 328 с.

17. Зеркалов Д.В. Безпека життєдіяльності. Навчальний посібник. К.: Основа. 2011. 526 с.

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

					<b>КС КРБ 123.115.00.00 ПЗ</b>	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А.  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“ \_\_\_ ” \_\_\_\_\_ 2024 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА УПРАВЛІННЯ СИЛОВИМИ РОЗЕТКАМИ

**ТЕХНІЧНЕ ЗАВДАННЯ**

на  8  листках

**Вид робіт:**

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

\_\_\_\_\_ к.т.н., доц. Жаровський Р.О.

\_\_\_\_\_ Головатий А. Р.

« \_\_\_ » \_\_\_\_\_ 2024 р.

« \_\_\_ » \_\_\_\_\_ 2024 р.

**Тернопіль 2024**



## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система управління силовими розетками».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.115.00.00

### 1.2 Виконавець

Студентка групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Головатий Артур Романович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету від « 24 » квітня 2024 року № 4/7-408

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 01.02.2024 р.

Плановий термін завершення виконання кваліфікаційної роботи – 24.06.2024 р.

### 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90%, наявності пояснювальної записки та графічного матеріалу.

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

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Система що розробляється призначена для:

- Індикації роботи силових розеток.
- Контролю за роботою силових розеток з допомогою таймера.
- Зберігання датих про пристрою.

### 2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка комп'ютеризованої системи, яка здійснює керування силовими розетками.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

#### 3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система повинна складатись з:

- мікроконтроллера;

- плати керування;
- додатку з графічним інтерфейсом.

### 3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість.

### 3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим функціонування;
- аварійний режим функціонування.

Для забезпечення нормального режиму функціонування системи необхідно виконувати вимоги і дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах (технічна документація, інструкції з експлуатації і т. д.).

Аварійний режим функціонування системи характеризується відмовою одного або декількох компонент програмного і (або) технічного забезпечення. При цьому функції роботи системи продовжують підтримувати роботу системи відстеження в межах базових налаштувань.

### 3.1.4 Вимоги по діагностуванню системи

Для діагностування системи відстеження використовуються інструменти діагностування основних процесів системи, які дозволяють перевірити основні функціональні елементи системи.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

### 3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки додаванню нових функцій і модулів.

## 3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

### 3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

Для захисту апаратури від стрибків напруги і комутаційних завад повинні застосовуватися мережні фільтри.

## 3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби мати занулення або захисне заземлення.

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання. У разі пожежі не має виділятися отруйних газів і димів. Після зняття електроживлення має бути доступне застосування будь-яких засобів пожежогасіння.

### 3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск  $760 \pm 25$  мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

### 3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії 1Д по класифікації документа, що діє, "Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем".

Компоненти підсистеми захисту від НСД повинні забезпечувати:

- ідентифікацію користувача;
- перевірку повноважень користувача при роботі з системою;
- розмежування доступу користувачів.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації,

повинен відповідати вимогам класу захищеності згідно вимогам документу “Засоби обчислювальної техніки. Захист від несанкціонованого доступу до інформації. Показники захищеності від несанкціонованого доступу до інформації”.

#### 3.4.1 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

### 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
  - Функціональна схема системи.
  - Схема електрична принципова.
  - Макет і локальний інтерфейс
  - Веб-інтерфейс налаштування системи.

\*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	01.02-09.02.2024
2	Аналіз технічного завдання	05.02-11.02.2024
3	Визначення і аналіз вимог до комп'ютеризованої системи	25.04-03.05.2024
4	Проектування структури комп'ютеризованої системи	04.05-20.05.2024
5	Проектування програмної частини	21.05-28.05.2024
6	Проектування апаратної частини	29.05-06.06.2024
7	Налаштування і тестування комп'ютеризованої системи	07.06-10.06.2024
8	Безпека життєдіяльності, основи охорони праці	10.06-15.06.2024
9	Оформлення кваліфікаційної роботи	16.06-20.06.2024
10	Попередній захист кваліфікаційної роботи	14.06.2024
11	Захист кваліфікаційної роботи	24.06-28.06.2024

## 6 Додаткові умови виконання кваліфікаційної роботи

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

## Додаток Б.

### Лістинг програмного забезпечення

```
#include < LiquidCrystal . h >
#include <EEPROM.h>
#include <MATH.h>
#include <EtherCard.h>
LiquidCrystal lcd (12, 11, 9, 8, 7, 6, 5, 4, 3, 2);
int rtcM=0, rtcH=0, rtcS=0; //Початкові значення змінних
часу int plset=0, p2set=0, p3set=0, p4set=0, t1set=0, t2set=0,
t3set=0, t4set=0;
//Початкові значення установок таймерів
int timer1=0, timer2=0, timer3=0, timer4=0, period1=0,
period2=0, period3=0, period4=0; //лічильники таймерів
int D=0, Dig=1, Sel=0, Dx=0; //Початковий екран
Display(0..4), перша цифра
bool b=0; //змінна відображення інформації на екрані
bool t1=0, t2=0, t3=0, t4=0; //Змінні стани таймерів
void setup () {
    lcd . begin (16, 2); // ініціалізуємо дисплей: 2 рядки по
16 символів readSet (); // зчитуємо раніше збережені
налаштування pinMode (22, OUTPUT ); // Призначення порту 22 на
вихід управління навантаженням таймера 1 pinMode (24, OUTPUT
); // Призначення порту 24 на вихід управління навантаженням
таймера 2 pinMode (26, OUTPUT ); // Призначення порту 26 на
вихід управління навантаженням таймера 3 pinMode (28, OUTPUT
); // Призначення порту 28 на вихід управління навантаженням
таймера 4
}
void loop () {
    xKey (); //запуск обробки натискань клавіш
timersWork (); //запуск обробки таймерів
setOut (); //Установка рівнів виходів таймерів }
void xKey () {
    int uKey = analogRead (0); // Створюємо змінну uKey та
опитуємо аналоговий порт А0, до якого підключені кнопки
    if ( uKey > 50) { // відсікаємо шум, якщо натиснута кнопка
if ( uKey < 100) { // Якщо uKey менше 100 натиснута кнопка
    "Вправо" Right
        if ( Sel <1) { // Перехід до наступного таймера
if ( D <5) D ++; else D = 0;
        }
        else { // Перехід до наступної цифри
if ( Dig <5) Dig ++; else Dig = 1;
        }
    }
    else if ( uKey < 200) { // Якщо uKey менше 200 натиснута
кнопка
    "Вгору" Up , то в залежності від виду подання
        if ( D ==0){ // основне вікно, відображаються 4
таймери, перехід до відображення поточного часу
```



```

if ( Sel > 0) { if ( Dx <10) Dx ++; else Dx = 0;
    }
else if(Dx<1) Dx++; else Dx=0;}
else { // таймер 1-4
if(Sel==0) { if(Dx<3) Dx++; else Dx=0;
}
else if(Sel==1) {
if(Dx<1441) Dx++; else Dx=1;
} else {
if(Dx<31) Dx++; else Dx=1;
}
}
}
else if (uKey < 400){ // Якщо uKey менше 400 натиснута
кнопка "Вниз" Down , то залежно від виду подання
    if ( D ==0) { // основне вікно, відображаються 4
таймери, перехід до відображення поточного часу
        if ( Sel >0) { if ( Dx >0) Dx --; else Dx = 9;
            }
        else if(Dx>0) Dx--; else Dx=1;
        }
    else { // таймер 1-4
if(Sel==0) {
if(Dx>0) Dx--;
else Dx=2;
}
else if(Sel==1) { if(Dx>1) Dx--; else Dx=1440;} else
{if(Dx>1) Dx--; else Dx=30;
}
}
}
else if (uKey < 600){ // Якщо uKey менше 600 натиснута
кнопка "Вліво" Left
    if(D>0) D--; else D=4; // Перехід до попереднього таймера чи
цифри
    }
else if (uKey < 800){ // Якщо uKey менше 800 натиснута
кнопка "Вибір" SELECT
    if ( D >0){ // Перемикання режиму відображення інформації
при кожному натисканні на кнопку по колу
        if(Sel<5) Sel++; else Sel=0;
    } else { if(Sel<2) Sel++; else Sel=0;
        }
        saveSet (); // Збереження введених даних щодо
натискання кнопки SELECT
    } }
    setPlay (); // Виведення інформації на екран }
void timersWork () {
    unsigned long tX = 0; //Змінна поточного часу
    if ( millis ()- tX >=60000){ //якщо кількість мілісекунд
стане більше 60сек, то
        b =! b ; //інверсія відображення значення змінної
відображення

```

```

if ( rtcM <60) { // поки кількість хвилин ще не 60
rtcM ++; //збільшуємо хвилини на одиницю
} else { //а інакше
rtcM = 0; //обнулюємо кількість хвилин
if ( rtcH <24) { // поки кількість годин ще не 24
rtcH ++; //збільшуємо годинник на одиницю
} else {
rtcH = 0; //а інакше обнуляємо кількість годин
if ( rtcS <31) { // поки кількість ще не 30
rtcS ++; //збільшуємо добу на одиницю
if ( period 1 <= p 1 set )
period 1++;
else period 1 = 0; //лічильник періоду таймера 1
if ( period 2 <= p 2 set ) period 2++; else period 2 = 0;
//лічильник періоду таймера 2
if ( period 3 <= p 3 set ) period 3++;
else period 3 = 0; //лічильник періоду таймера 3
if ( period 4 <= p 4 set )
period 4++;
else period 4 = 0; //лічильник періоду таймера 4
} else { rtcS = 0; //а інакше обнуляємо добу
}
}
}
if(timer1 <= t1set) timer1++; else timer1 = 0;
//лічильник таймера 1
if(timer2 <= t2set) timer2++; else timer2=0;
//лічильник таймера 2
if(timer3 <= t3set) timer3++; else timer3=0;
//лічильник таймера 3
if(timer4 <= t4set) timer4++; else timer4=0;
//лічильник таймера 4
do { //в циклі захисту від переповнення таймерів
мікроконтролера
tX + = 60000; //збільшуємо змінну часу на 60 секунд
if ( tX <60000) break ; //якщо змінна стане менше 60
секунд, тобто. лічильник переповниться, виходимо з циклу
}
while ( tX < millis ()-60000); //доки змінна часу менше ніж
різниця між поточним значенням функції часу і 60 секундами
} }
void setPlay () {
if ( Sel <1) { // основні екрани Sel = 0
if(D<1){ // D=0 if(Dx<1) screen1(); // Основний екран
Dx=0
else screenW(); // Годинник Dx = 1
}
else if(Dx<1) screen2(); // Таймери D=1-4, Dx=0 else
if(Dx>1) screen4(); // Dx=2 else screen3(); // Dx = 1
} // екрани установок Sel=1,2 else if(D<1) screenSet1();
// установка годинника D=0
else if(Sel>1) screenSet3(); // встановлення тривалості
таймера Sel=2

```

```

else screenSet2(); // Установка періодичності таймера
Sel = 1 }
void screen 1(){ //основний екран із 4 таймерами
lcd . clear (); //Очищення буфера екрану
    lcd . setCursor (0, 0); //Встановити курсор на перший
рядок і перший символ
    String tr = ":"; //Визначити відображення роздільника годин
і хвилин
    if ( b ) tr = ""; //Визначити миготіння змінної - стирання
відображення на поточний прохід
    int rtcH 1 = timer 1/60; //Визначити кількість годин
таймера 1
    int rtcH 2 = timer 2/60; //Визначити кількість годин таймера
2
    int rtcH 3 = timer 3/60; //Визначити кількість годин таймера
3
    int rtcH 4 = timer 4/60; //Визначити кількість годин таймера
4
    int rtcM 1 = timer 1 - rtcH 1 * 60; //Визначити кількість
хвилин таймера 1
    int rtcM 2 = timer 1 - rtcH 1 * 60; //Визначити кількість
хвилин таймера 2
    int rtcM 3 = timer 1 - rtcH 1 * 60; //Визначити кількість
хвилин таймера 3
    int rtcM 4 = timer 1 - rtcH 1 * 60; //Визначити кількість
хвилин таймера 4
    String rtcHs 1 = String ( rtcH 1); //Визначити відображення
змінної годинника таймера 1
    String rtcMs 1 = String ( rtcM 1); //Визначити
відображення змінної хвилин таймера 1
    String rtcHs 2 = String ( rtcH 2); //Визначити
відображення змінної годинника таймера 2
    String rtcMs 2 = String ( rtcM 2); //Визначити
відображення змінної хвилин таймера 2
    String rtcHs 3 = String ( rtcH 3); //Визначити
відображення змінної годинника таймера 3
    String rtcMs 3 = String ( rtcM 3); //Визначити
відображення змінної хвилин таймера 3
    String rtcHs 4 = String ( rtcH 4); //Визначити
відображення змінної годинника таймера 4
    String rtcMs 4 = String ( rtcM 4); //Визначити
відображення змінної хвилин таймера 4
    String tmNs1 = String(t1?"*":" "); //Визначити стан
таймера 1 0 = false (OFF), 1 = true (ON)
    String tmNs2 = String(t2?"*":" "); //Визначити стан таймера
2 0 = false (OFF), 1 = true (ON)
    String tmNs3 = String(t3?"*":" "); //Визначити стан таймера
3 0 = false (OFF), 1 = true (ON)
    String tmNs4 = String(t4?"*":" "); //Визначити стан таймера
4 0 = false (OFF), 1 = true (ON)
    String Firststr = String(" " + rtcHs1 + ":" + rtcMs1 + " " +
tmNs1 + " " + rtcHs3 +
":" + rtcMs 3 + " " + tmNs 3); //формування першого рядка

```

```

        lcd . println ( Firststr ); //Виведення на екран першого
рядка
        String Secondstr = String ( " " + rtcHs 2 + ":" + rtcMs 2 + "
" + tmNs 2 + " " + rtcHs 4 + ":" + rtcMs 4 + " " + tmNs 4);
        //формування другого рядка
        lcd . println ( Secondstr ); //Виведення на екран другого
рядка }

void screenW () { //екран відображення поточного часу Watch
        lcd . clear (); //Очищення буфера екрану
        lcd . setCursor ( 0, 0); //Встановити курсор на перший
рядок і перший символ
        String tr = ":"; //Визначити відображення роздільника годин
і хвилин
        if ( b ) tr = ""; //Визначити миготіння змінної - стирання
відображення на поточний прохід
        String rtcHs = String ( rtcH ); //Визначити відображення
змінної годинника
        String rtcMs = String ( rtcM ); //Визначити відображення
змінної хвилин
        String Firststr = String ( " Now time " + rtcHs + tr + rtcMs
); // формування першого рядка
        lcd . println ( Firststr ); //Виведення на екран першого
рядка
        String Secondstr = String ( " Uptime " + String ( rtcS ) + "
day "); //формування другого рядка
        lcd . println ( Secondstr ); //Виведення на екран другого
рядка
    } void screen 2() { //екран таймера lcd . clear (); //
Чищення буфера екрану
        lcd . setCursor ( 0, 0); //Встановити курсор на перший
рядок і перший символ
        String nT = String ( D ); //Номер таймера рівний номеру
дисплея 1..4
        String tmNs = ""; //Визначити відображення змінної годинника
таймера switch ( D ) {
            // визначити стан таймера 0 = false ( OFF ), 1 = true ( ON )
            case 1: tmNs = String ( t 1? " ON ": " OFF "); break ;
            case 2: tmNs = String ( t 2? " ON ": " OFF "); break ;
            case 3: tmNs = String ( t 3? " ON ": " OFF "); break ;
            case 4: tmNs = String ( t 4? " ON ": " OFF "); break ;
        }
        String Firststr = String ( " T " + nT + " " + tmNs + " m ");
//формування першого рядка з номера та поточного часу таймера
        lcd.println(Firststr); //виведення на екран першого рядка
        String Secondstr = String("Period" + String(plset) + "day");
//формування
        другого рядка із встановленою величиною періодичності
включення таймера
        lcd . println ( Secondstr ); //Виведення на екран другого
рядка
    }
    void screen 3() { //екран минулого часу таймера

```

```

        lcd . clear (); //Очищення буфера екрану
        lcd . setCursor (0, 0); //Встановити курсор на перший
рядок і перший символ
        String tr = ":"; //Визначити відображення роздільника годин
і хвилин
        int tmH = 0, tmM = 0, tmE = 0; //Визначити змінні годин і
хвилин таймера switch ( D ) { // обчислити змінні годин і
хвилин залежно від
номери таймера
case 1: tmE = timer1; break;
case 2: tmE = timer2; break;
case 3: tmE = timer3; break;
case 4: tmE = timer4; break;
}
        tmH = tmE/60; tmM = tmE - tmH * 60;
        String tmHs = String ( tmH ); //Визначити відображення
змінної годинника
таймера
        String tmMs = String ( tmM ); //Визначити відображення
змінної хвилин таймера
        if ( b ) tr = ""; //Визначити миготіння змінної - стирання
відображення на поточний прохід
        String tmNs = ""; //визначити відображення змінної
годинника таймера switch(D){ //визначити стан таймера 0=false
(OFF), 1=true (ON)
        case 1: tmNs = String(t1?"ON":"OFF"); break;
        case 2: tmNs = String(t2?"ON":"OFF"); break;
        case 3: tmNs = String(t3?"ON":"OFF"); break;
        case 4: tmNs = String(t4?"ON":"OFF"); break;
        }
        String Firststr = String ( " T " + String ( D ) + " " + tmNs
); //формування першого рядка з номера таймера та його стану
        lcd . println ( Firststr ); //Виведення на екран першого
рядка
        String Secondstr = String("PASTtime" + tmHs + tr + tmMs);
//формування другого рядка з часу, що минуло з моменту
включення таймера
        lcd . println ( Secondstr ); //Виведення на екран другого
рядка }

void screen 4(){ //екран часу, що залишився таймера
        String tr = ":"; //Визначити відображення роздільника
годин і хвилин
        int tmH = 0, tmM = 0, tmE = 0; //Визначити змінні годин і
хвилин таймера
        switch ( D ) { // обчислити змінні годин і хвилин залежно
від номера таймера
                case 1: tmE = t1set - timer1; break;
                case 2: tmE = t2set - timer2; break;
                case 3: tmE = t3set - timer3; break;
                case 4: tmE = t4set - timer4; break;
        }
        tmH = tmE/60;

```

```

tmM = tmE - tmH * 60;
String tmHs = String ( tmH ); //Визначити відображення
змінної годинника таймера
String tmMs = String ( tmM ); //Визначити відображення
змінної хвилин таймера
if ( b ) tr = ""; //Визначити миготіння змінної - стирання
відображення на поточний прохід
String tmNs = ""; //Визначити відображення змінної
годинника таймера
switch ( D ) { // визначити стан таймера 0 = false ( OFF ),
1 = true ( ON )
case 1:
tmNs = String(t1?"ON":"OFF"); break; case 2:
tmNs = String(t2?"ON":"OFF"); break; case 3:
tmNs = String(t3?"ON":"OFF"); break; case 4:
tmNs = String(t4?"ON":"OFF"); break;
}
String Firststr = String ( " T " + String ( D ) + " " + tmNs
); //формування першого рядка з номера таймера та його стану
lcd . println ( Firststr ); //Виведення на екран першого
рядка
String Secondstr = String("PASTtime" + tmHs + tr + tmMs);
//формування другого рядка з часу, що минув з моменту
включення таймера
lcd . println ( Secondstr ); //Виведення на екран другого
рядка
}
void screenSet 1(){ //Екран
установки поточного часу Watch
lcd . clear (); //Очищення буфера екрану
lcd . setCursor (0, 0); //Встановити курсор на перший
рядок і перший символ String Firststr = String ( " Set now time
"); // формування першого рядка lcd . println ( Firststr );
//Виведення на екран першого рядка
String Secondstr , tmStr ; //Визначити змінні другого
рядка int tmDH = 0, tmEH = 0, tmDM = 0, tmEM = 0; //Визначити
змінні цифр годин і хвилин tmDH = rtcH /10; tmEH = rtcH - tmDH
* 10; tmDM = rtcM /10; tmEM = rtcM - tmDM * 10;
//Визначити миготіння змінної - стирання відображення на
поточний прохід
switch ( D ) { case 1:
if(b) tmStr = ""; else tmStr = String(tmDH);
Secondstr = tmStr + String(tmEH) + ":" + String(tmDM) +
String(tmEM); break; case 2:
if(b) tmStr = ""; else tmStr = String(tmEH);
Secondstr = String(tmDH) + tmStr + ":" + String(tmDM) +
String(tmEM); break; case 3:
if(b) tmStr = ""; else tmStr = String(tmDM);
Secondstr = String(tmDH) + String(tmEH) + ":" + tmStr +
String(tmEM); break; case 4:
if(b) tmStr = ""; else tmStr = String(tmEM);
Secondstr = String(tmDH) + String(tmEH) + ":" + String(tmDM)
+ tmStr; break;

```

```

    }
    lcd . println ( Secondstr ); //Виведення на екран другого
рядка
    rtcH = tmDH * 10 + tmEH ; //збереження змінених даних у
змінну годинника
    rtcM = tmDM * 10 + tmEM ; //збереження змінених даних у
змінну хвилини
    }
    void screenSet 2(){ //екран налаштування тривалості
включення таймера
        lcd . clear (); //Очищення буфера екрану
        lcd . setCursor (0, 0); //Встановити курсор на перший
рядок і перший символ
        String Firststr = String (" Set timer " + String ( D )); //
формування першого рядка
        lcd . println ( Firststr ); //Виведення на екран першого
рядка
        String Secondstr , tmStr ; //Визначити змінні другого рядка
if ( b ) tmStr = ""; else tmStr = String ( Dx ); //Визначити
миготіння змінної - стирання відображення на поточний прохід
        Secondstr = " ON time " + tmStr + " m "; // формування
другого рядка
        lcd . println ( Secondstr ); //Виведення на екран другого
рядка
        //збереження змінених даних у змінну відповідного
таймера
        switch(D){ case 1: timer1 = Dx; break; case 2: timer2 = Dx;
break; case 3: timer3 = Dx; break; case 4: timer4 = Dx; break; }
    }
    void screenSet 3(){ //екран налаштування періодичності
включення таймера
        lcd . clear (); //Очищення буфера екрану
        lcd . setCursor (0, 0); //Встановити курсор на першу
рядок та перший символ
        String Firststr = String (" Set timer " + String ( D ));
// формування першого рядка
        lcd . println ( Firststr ); //Виведення на екран першого
рядка
        String Secondstr , tmStr
; //Визначити змінні другого рядка
        if ( b ) tmStr = ""; else tmStr = String ( Dx );
//Визначити миготіння змінної - стирання відображення на
поточний прохід
        Secondstr = " Period " + tmStr + " day "; //формування
другого рядка lcd . println ( Secondstr ); //Виведення на
екран другого рядка
        //збереження змінених даних у змінну відповідного
таймера
        switch (D) {
        case 1: period1 = Dx; break;
        case 2: period2 = Dx; break;
        case 3: period3 = Dx; break;
        case 4: period4 = Dx; break;

```

```

    }
}
void setOut () { // встановлення рівня виходу
    if ( timer 1>0) {
        digitalWrite (22, HIGH ); t 1=1;} //якщо час таймера 1
        більше 0 (включено) включити вихід по 22 порту
    } else {
        digitalWrite (22, LOW ); t 1=0;} //якщо час таймера 1
        дорівнює 0 (вимкнений) вимкнути вихід по 22 порту
    } if ( timer 2>0) {
        digitalWrite (24, HIGH ); t 2=1;} //якщо час таймера 2
        більше 0 (включено) включити вихід по 24 порту
    } else { digitalWrite (24, LOW ); t 2=0;} //якщо час таймера 2
        дорівнює 0 (вимкнений) вимкнути вихід по 24 порту
    } if ( timer 3>0) {
        digitalWrite (26, HIGH ); t 3=1;} //якщо час таймера 3
        більше 0 (включено) включити вихід по 26 порту
    } else { digitalWrite (26, LOW ); t 3=0;}
        //якщо час таймера 3 дорівнює 0 (вимкнено) вимкнути вихід по
        26 порту
    } if ( timer 4>0) { digitalWrite (28, HIGH ); t 4=1;}
        //якщо час таймера 4 більше 0 (включено) включити вихід по
        28 порту
    } else { digitalWrite (28, LOW ); t 4=0;}
        //якщо час таймера 4 дорівнює 0 (вимкнено) вимкнути вихід по
        28 порту
    }
}
void saveSet ()
{
EEPROM . put (0, t 1 set ); //Оновлення дл-ти таймера 1 ( int
займає 2 байти)
EEPROM . put (3, t 2 set ); // Оновлення тривалості таймера 2
EEPROM . put (5, t 3 set ); // Оновлення тривалості таймера 3
EEPROM . put (7, t 4 set ); // Оновлення тривалості таймера 4
EEPROM . put (9, p 1 set ); // Оновлення періодичності таймера 1
EEPROM . put (11, p 2 set ); // Оновлення періодичності таймера
2
EEPROM . put (13, p 3 set ); // Оновлення періодичності таймера
3
EEPROM . put (15, p 4 set ); // Оновлення періодичності таймера
4
}
void readSet ()
{
EEPROM . get (0, t 1 set ); // Читання тривалості таймера 1
EEPROM . get (3, t 2 set ); // Читання тривалості таймера 2
EEPROM . get (5, t 3 set ); // Читання тривалості таймера 3
EEPROM . get (7, t 4 set ); // Читання тривалості таймера 4
EEPROM . get (9, p 1 set ); // Читання періодичності таймера 1
EEPROM . get (11, p 2 set ); // Читання періодичності таймера 2
EEPROM . get (13, p 3 set ); // Читання періодичності таймера 3
EEPROM . get (15, p 4 set ); // Читання періодичності таймера 4
}
}

```