

АНОТАЦІЯ

Метою цієї кваліфікаційної роботи є розробка та реалізація системи контролю та керування енергоспоживанням. У роботі досліджуються сучасні засоби контролю енергоспоживання. Створення системи контролю та керування енергоспоживанням на базі мікроконтролерної плати Arduino Nano, датчика струму ACS712, рідкокристалічного дисплея LCD 1602 I2C та електромеханічного. Цей проект демонструє, як за допомогою доступних та відносно недорогих компонентів можна реалізувати функціональне рішення для моніторингу та базового управління електричними навантаженнями. Застосування програмних методів згладжування ("ковзне середнє") є необхідним для мінімізації впливу електричного шуму. Розроблена система може знайти застосування в таких сферах, як домашній моніторинг енергоспоживання, моніторинг у малих офісах/майстернях, освітні та демонстраційні цілі, як основа для прототипування.

ABSTRACT

The purpose of this qualification work is to develop and implement a system for monitoring and controlling energy consumption. The work investigates modern means of monitoring energy consumption. Creation of a system for monitoring and controlling energy consumption based on the Arduino Nano microcontroller board, the ACS712 current sensor, the LCD 1602 I2C liquid crystal display and electromechanical. This project demonstrates how, using affordable and relatively inexpensive components, a functional solution for monitoring and basic control of electrical loads can be implemented. The use of software smoothing methods ("moving average") is necessary to minimize the impact of electrical noise. The developed system can be used in such areas as home energy consumption monitoring, monitoring in small offices/workshops, educational and demonstration purposes, as a basis for prototyping.

ЗМІСТ

ВСТУП.....	6
1. ЗБІР ДАНИХ ПРО ЕНЕРГОСПОЖИВАННЯ ДЛЯ СИСТЕМ КОНТРОЛЮ	8
1.1 Визначення джерел даних про енергоспоживання.....	8
1.2 Типи джерел даних та методи їх отримання за допомогою мікроконтролерів	9
1.3 Вибір сенсорів для вимірювання енергоспоживання	12
1.4 Обробка даних на мікроконтролері.....	16
2 ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ КОНТРОЛЮ ТА КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ.....	21
2.1 Вибір мікроконтролера.....	21
2.1.1 Arduino (на базі ATmega328P).....	21
2.1.2 ESP32.....	24
2.1.3 STM32	27
2.2 Вибір сенсорів струму для системи контролю енергоспоживання..	31
2.2.1 Сенсор струму ACS712.....	32
2.2.2 Сенсор струму SCT-013-000.....	35
2.2.3 Сенсора струму та напруги INA219	37
2.3 Вибір засобу відображення інформації.....	40
2.3.1 Семисегментні індикатори	40
2.3.2 Рідкокристалічні дисплеї (LCD) на основі HD44780.....	41
2.3.3 OLED-дисплеї	42
2.3.4 Графічні LCD-дисплеї (GLCD)	43
2.3.5 TFT LCD-дисплеї.....	44
2.3.6 Передача даних на комп'ютер або смартфон.....	45
2.4 Вибір засобів керування енергоспоживанням.....	46
2.4.1 Реле (Electromechanical Relays).....	46

2.4.2 Твердотільні реле (Solid State Relays - SSR)	47
2.4.3 Симістори та триаки.....	47
2.4.4 Транзистори (MOSFET, ВJT)	48
3 ПРОЕКТУВАННЯ СИСТЕМИ КОНТРОЛЮ ТА КЕРУВАННЯ	
ЕНЕРГОСПОЖИВАННЯМ.....	50
3.1 Апаратна реалізація системи	50
3.2 Програмна реалізація.....	55
ВИСНОВКИ.....	62
ДОДАТОК А.....	67

ВСТУП

В умовах стрімкого зростання світового населення, індустріалізації та технологічного прогресу, питання ефективного та раціонального використання енергетичних ресурсів набуває першочергового значення. З одного боку, зростаюча потреба в енергії чинить значний тиск на існуючі енергетичні системи, призводячи до виснаження природних ресурсів та посилення негативного впливу на навколишнє середовище. З іншого боку, неоптимізоване споживання енергії призводить до значних фінансових витрат як на рівні окремих домогосподарств, так і на рівні промислових підприємств та цілих держав [1,2].

У цьому контексті, розробка та впровадження інтелектуальних систем контролю та керування енергоспоживанням є ключовим напрямком для досягнення енергетичної ефективності та сталого розвитку. Такі системи здатні здійснювати моніторинг енергетичних потоків у реальному часі, аналізувати отримані дані, виявляти неефективні режими роботи обладнання та автоматично або за допомогою рекомендацій оптимізувати споживання енергії.

Актуальність даної роботи зумовлена, насамперед, економічною доцільністю. Зниження витрат на енергоносії є одним з найважливіших стимулів для впровадження систем контролю та керування енергоспоживанням. Оптимізація використання енергії дозволяє зменшити операційні витрати підприємств та комунальні платежі населення.

Раціональне використання енергії сприяє зменшенню викидів парникових газів та інших забруднюючих речовин, що є критично важливим для боротьби зі зміною клімату та збереження навколишнього середовища. Системи моніторингу енергоспоживання можуть виявляти аномалії в роботі обладнання, запобігаючи аваріям та забезпечуючи більш стабільне енергопостачання. Сучасні досягнення в галузі мікроелектроніки, сенсорних технологій, бездротових комунікацій та штучного інтелекту створюють

широкі можливості для розробки ефективних та доступних систем контролю та керування енергоспоживанням.

Практичне значення даної роботи полягає у можливості створення недорогої та гнучкої системи контролю та керування енергоспоживанням. Розроблена система може бути використана для моніторингу енергоспоживання в домогосподарствах, офісах, невеликих підприємствах, а також для експериментальних досліджень в галузі енергоефективності. Отримані результати можуть бути використані для подальшого вдосконалення існуючих систем та розробки нових, більш ефективних рішень у сфері енергозбереження.

Таким чином, дана робота є актуальним та важливим дослідженням, спрямованим на розробку практично значущої системи контролю та керування енергоспоживанням з використанням доступних та гнучких технологій.

1. ЗБІР ДАНИХ ПРО ЕНЕРГОСПОЖИВАННЯ ДЛЯ СИСТЕМ КОНТРОЛЮ

Побудова ефективної системи контролю енергоспоживання на починається з ретельного збору даних. Якість і повнота цих даних безпосередньо впливають на точність аналізу, ефективність прийнятих рішень та загальну функціональність системи. Цей процес включає визначення джерел даних, вибір відповідних сенсорів та методів їх підключення, розробку алгоритмів обробки отриманих значень та забезпечення їхньої надійності [3].

1.1 Визначення джерел даних про енергоспоживання

Першим кроком є чітке визначення об'єктів, енергоспоживання яких необхідно контролювати. Це можуть бути окремі побутові прилади (холодильник, пральна машина, освітлення), групи приладів, окремі розетки або навіть загальне споживання електроенергії в будинку чи квартирі. Вибір джерел даних залежить від мети системи контролю: чи це моніторинг споживання окремих приладів для виявлення неефективних з них, чи загальний облік для оптимізації використання енергії.

На цьому етапі важливо також визначити необхідний рівень деталізації даних. Чи достатньо знати загальне споживання за певний період часу (наприклад, за годину, день), чи потрібні дані в реальному часі для миттєвого реагування на зміни? Відповідь на це питання вплине на вибір сенсорів та частоту опитування.

Для побудови ефективної системи контролю енергоспоживання, першочерговим кроком є чітке визначення джерел даних, які будуть використовуватися для моніторингу та аналізу. Від правильного вибору цих джерел залежить повнота інформації про енергетичні потоки, точність прийнятих рішень щодо оптимізації та загальна функціональність системи. Обсяг та деталізація необхідних даних визначаються метою системи

контролю: чи це моніторинг окремих приладів для виявлення неефективних з них, чи загальний облік для оптимізації споживання на рівні домогосподарства або підприємства.

Визначення джерел даних слід починати з розуміння необхідного рівня агрегації інформації. Дані про загальне споживання електроенергії об'єктом (будинком, квартирою, цехом). Джерелом таких даних зазвичай є основний лічильник електроенергії, встановлений постачальником послуг. Дані про споживання енергії певними групами приладів, об'єднаних за функціональною ознакою (освітлення, опалення, кухонне обладнання) або за розташуванням (розетки в певній кімнаті). Це дозволяє аналізувати енергоефективність різних систем об'єкта та виявляти найбільш енергоємні групи. Джерелами можуть бути окремі лічильники або датчики, встановлені на відповідних електричних колах. Детальні дані про енергоспоживання кожного окремого приладу (холодильника, пральної машини, комп'ютера). Це забезпечує найвищий рівень деталізації та дозволяє точно визначити енергоефективність кожного пристрою, виявити несправності або неефективні режими роботи. Джерелами можуть бути окремі розеткові лічильники, смарт-розетки або датчики, інтегровані безпосередньо в прилади (за наявності такої функціональності).

Вибір рівня агрегації залежить від конкретних завдань системи контролю. Для загальної оптимізації може бути достатньо даних про загальне споживання та споживання основних груп приладів. Для детального аналізу енергоефективності та виявлення проблемних пристроїв необхідно збирати дані на рівні окремих приладів.

1.2 Типи джерел даних та методи їх отримання за допомогою мікроконтролерів

Після визначення необхідного рівня агрегації даних, слід обрати конкретні джерела даних та методи їх отримання за допомогою

мікроконтролеру.

Багато сучасних лічильників електроенергії мають імпульсний вихід, де кожен імпульс відповідає певній кількості спожитої енергії (наприклад, 1 імпульс = 0.001 кВт/год). Мікроконтролер може підраховувати ці імпульси за допомогою цифрового входу. Частота імпульсів пропорційна миттєвій потужності. Деякі лічильники можуть мати цифрові інтерфейси зв'язку, такі як RS-485 (протокол Modbus) або M-Bus. Для роботи з ними контролер вимагає відповідного модуля інтерфейсу та програмної реалізації протоколу. Це складніший підхід, але може забезпечити більш детальну інформацію. Для неінвазивного моніторингу загального струму, що проходить через основний кабель живлення, можна використовувати потужні трансформатори струму, підключені до аналогових входів мікроконтролеру через навантажувальний резистор та схему зсуву напруги. У поєднанні з інформацією про напругу в мережі (яку можна вважати відносно стабільною або вимірювати окремим датчиком напруги), можна розрахувати загальну споживану потужність.

Для моніторингу споживання окремих груп приладів, підключених до окремих автоматичних вимикачів у розподільчому щиті, можна використовувати менші за розміром трансформаторів струму, встановлені на відповідних проводах. Для більш точного вимірювання струму в окремих колах можуть використовуватися резистивні шунти, але їх встановлення вимагає розриву електричного кола та забезпечує відсутність гальванічної ізоляції, що є менш безпечним варіантом для непрофесіоналів. Можна використовувати компактні однофазні або трифазні лічильники електроенергії з імпульсними або цифровими виходами, підключені до відповідних електричних кіл, а їхні виходи - до мікроконтролеру.

Для спрощення системи можна використовувати готові розеткові лічильники з вбудованими датчиками струму та напруги, які можуть мати інтерфейси зв'язку (наприклад, UART, Wi-Fi) для передачі даних на мікроконтролер або безпосередньо в мережу. Крім того є розумні розетки з

функцією моніторингу енергоспоживання, які можуть передавати дані через Wi-Fi за протоколами, такими як MQTT або HTTP, які мікроконтролер може обробляти за наявності відповідного модуля та бібліотеки. Деякі сучасні "розумні" побутові прилади можуть надавати дані про своє енергоспоживання через API (інтерфейс програмування додатків) у локальній мережі або через хмарні сервіси. Для отримання цих даних мікроконтролер повинен бути підключений до мережі та мати можливість здійснювати відповідні запити. Для моніторингу струму, що споживається окремим приладом, можна використовувати невеликі трансформатори струму, встановлені на його шнурі живлення (слід охоплювати лише один провідник-фазу або нуль).

Окрім безпосередніх вимірювань електричного струму та напруги, для більш повної картини енергоспоживання та контекстного керування можуть використовуватися й інші джерела даних.

Для автоматичного керування освітленням залежно від рівня природного світла – датчики освітленості. Для виявлення присутності людей та керування освітленням або іншими приладами залежно від цього – датчики руху. Для моніторингу кліматичних умов та оптимізації роботи систем опалення та кондиціонування – датчики температури та вологості. Дані про поточні тарифи на електроенергію (наприклад, з веб-сервісів) можуть використовуватися для прийняття рішень про час використання енергоємних приладів. Прогнози сонячної активності можуть бути корисними для систем з використанням сонячних панелей.

При виборі джерел даних для системи контролю енергоспоживання на мікроконтролеї слід враховувати ряд факторів. Різні типи датчиків мають різну точність. Вибір залежить від вимог до точності обліку та керування. Вартість датчиків та необхідних модулів може варіюватися. Слід враховувати бюджет проекту. Неінвазивні датчики струму є простішими та безпечнішими у встановленні порівняно з шунтами, які вимагають розриву

кола. Робота з високовольтними мережами потребує особливої кваліфікації та дотримання правил безпеки. Датчики повинні мати інтерфейс, сумісний з мікроконтролером (аналоговий або цифровий вихід, протоколи зв'язку). Обрані джерела даних повинні забезпечувати необхідний рівень деталізації для досягнення мети системи контролю. Якщо планується розширення системи в майбутньому, слід обирати джерела даних та методи, які легко масштабуються. Слід враховувати власне енергоспоживання компонентів системи контролю.

Визначення джерел даних є критично важливим етапом у розробці системи контролю енергоспоживання на мікроконтролері [4,5]. Ретельний аналіз потреб системи, розуміння доступних джерел даних та їхніх характеристик, а також врахування факторів точності, вартості та складності встановлення дозволить створити ефективну та інформативну систему, яка сприятиме оптимізації енергоспоживання та досягненню поставлених цілей. Подальша розробка системи включатиме вибір відповідних сенсорів, розробку алгоритмів збору та обробки даних, а також реалізацію логіки керування навантаженнями на основі отриманої інформації.

1.3 Вибір сенсорів для вимірювання енергоспоживання

Існує кілька типів сенсорів, які можуть бути використані для вимірювання енергоспоживання, кожен з яких має свої переваги та недоліки.

Сенсори струму (Current Sensors) – ці сенсори вимірюють силу струму, що протікає через провідник. Існує декілька основних типів:

Трансформатори струму (Current Transformers - CT) - працюють за принципом електромагнітної індукції. Змінний струм, що протікає через первинну обмотку (провідник, який необхідно виміряти), створює змінне магнітне поле, яке індукує змінний струм у вторинній обмотці. Величина струму у вторинній обмотці пропорційна струму в первинній обмотці з відомим коефіцієнтом трансформації.

Більшість СТ є неінвазивними, тобто вони не потребують розриву електричного кола для встановлення. Датчик просто охоплює провідник. Це забезпечує безпеку та зручність монтажу. СТ забезпечують гальванічну ізоляцію між вимірюваним колом високої напруги та низьковольтним колом мікроконтролера, що підвищує безпеку. СТ ідеально підходять для вимірювання змінного струму, який є типовим для побутових та промислових мереж. Доступні СТ з різними діапазонами вимірювання струму, від одиниць ампер до сотень і тисяч ампер. Багато моделей СТ є досить доступними за ціною.

Однак, СТ не можуть вимірювати постійний струм (DC). Для перетворення вихідного струму СТ в напругу, яку може зчитати мікроконтролер, необхідно використовувати навантажувальний резистор. Правильний вибір опору цього резистора є важливим для точності вимірювання та запобігання перенасичення трансформатора. Точність СТ може змінюватися залежно від величини вимірюваного струму та опору навантажувального резистора. СТ можуть вносити невеликий фазовий зсув між струмом та напругою, що може впливати на точність вимірювання активної потужності.

Резистивні шунти (Current Shunts) - резистивний шунт являє собою прецизійний резистор з дуже малим відомим опором, який включається послідовно в електричне коло, струм якого необхідно виміряти. Протікання струму через шунт викликає падіння напруги на ньому відповідно до закону Ома ($V=I \times R$). Вимірюючи цю напругу, можна визначити силу струму.

Шунти можуть використовуватися для вимірювання як постійного, так і змінного струму. При використанні прецизійних шунтів можна досягти високої точності вимірювання струму. Доступні шунти з різними номінальними струмами.

Однак, для встановлення шунта необхідно розірвати електричне коло та включити його послідовно, що може бути незручним та небезпечним, особливо у високовольтних системах. Шунт безпосередньо підключений до

вимірюваного кола, тому немає гальванічної ізоляції між високовольтним колом та мікроконтролером. Це вимагає особливої уважності до безпеки. Протікання струму через шунт призводить до невеликих втрат потужності у вигляді тепла ($P=I^2 \times R$). Для великих струмів ці втрати можуть бути значними. Падіння напруги на шунті часто є невеликим, тому може знадобитися підсилювач напруги для забезпечення достатнього рівня сигналу для АЦП мікроконтролера.

Датчики струму на основі ефекту Холла (Hall Effect Current Sensors) використовують явище виникнення поперечної різниці потенціалів (напруги Холла) в провіднику зі струмом, поміщеному в магнітне поле, перпендикулярне до напрямку струму. Магнітне поле зазвичай створюється струмом, що протікає через провідник, який необхідно виміряти, або через котушку з струмом.

Датчики Холла здатні вимірювати як постійний, так і змінний струм, деякі датчики Холла забезпечують гальванічну ізоляцію між вимірюваним колом та вихідним сигналом. Існують неінвазивні датчики Холла. Датчики Холла зазвичай мають лінійну залежність між вимірюваним струмом та вихідною напругою, що полегшує обробку даних.

Однак, датчики Холла, як правило, є дорогими порівняно з СТ та шунтами. Точність датчиків Холла може бути нижчою, особливо в порівнянні з прецизійними шунтами. Зовнішні магнітні поля можуть впливати на точність вимірювань.

Інтегровані датчики струму (Integrated Current Sensors) – компактні мікросхеми, які об'єднують сенсор струму (часто на основі ефекту Холла або магніторезистивного ефекту) та схему обробки сигналу. Вони зазвичай видають аналогову напругу або цифровий сигнал, пропорційний вимірюваному струму. Прикладом є серія ACS712.

Інтегровані датчики є невеликими за розміром та мають просте підключення до мікроконтролера, вони часто мають вбудовану схему підсилення та фільтрації сигналу. Деякі інтегровані датчики забезпечують

гальванічну ізоляцію. Більшість інтегрованих датчиків здатні вимірювати обидва типи струму.

Однак, інтегровані датчики зазвичай мають обмежений діапазон вимірюваних струмів (наприклад, $\pm 5\text{A}$, $\pm 20\text{A}$, $\pm 30\text{A}$ для ACS712). Точність може бути нижчою порівняно з більш спеціалізованими датчиками. Вихідний сигнал може бути чутливим до електричних шумів.

Докладне порівняння цих сенсорів для вимірювання енергоспоживання представлено в таблиці 1.1.

Таблиця 1.1 Порівняльна таблиця датчиків струму

Характеристика	Трансформатори струму (СТ)	Резистивні шунти	Датчики струму Холла	Інтегровані датчики струму
Принцип дії	Електромагнітна індукція	Закон Ома	Ефект Холла	Різні (часто Холла/магніторез)
Вимірювання DC	Ні	Так	Так	Так
Вимірювання AC	Так	Так	Так	Так
Гальванічна ізоляція	Так	Ні	Деякі моделі	Деякі моделі
Інвазивність	Неінвазивні	Інвазивні	Деякі неінвазивні	Зазвичай інвазивні
Точність	Середня	Висока	Середня	Середня

Сенсори напруги (Voltage Sensors) – вимірюють напругу в електричній мережі. Для більшості побутових мереж напруга є відносно стабільною, тому часто достатньо одноразового вимірювання або використання стандартного значення. Однак, для більш точного розрахунку потужності, особливо при змінному навантаженні, може знадобитися постійний моніторинг напруги.

Комбіновані сенсори потужності (Power Sensors/Energy Meters) – ці сенсори безпосередньо вимірюють активну потужність (у ватах) або енергію (у ват-годинах). Вони часто включають в себе вимірювання як струму, так і напруги та виконують необхідні розрахунки. Прикладами можуть бути готові

модулі лічильників електроенергії з інтерфейсами для підключення до мікроконтролерів.

При виборі сенсора необхідно враховувати наступні фактори:

- Сенсор повинен охоплювати очікуваний діапазон струму або потужності контролюваного приладу або мережі.
- Точність вимірювання є критично важливою для отримання достовірних даних.
- Безпека підключення та експлуатації сенсора є пріоритетом, особливо при роботі з високою напругою. Неінвазивні сенсори струму (СТ) є більш безпечними в цьому плані.
- Сенсор повинен мати інтерфейс, сумісний з контролером (наприклад, аналоговий або цифровий вихід).
- Вартість сенсора також є важливим фактором, особливо при побудові великої кількості точок моніторингу.

1.4 Обробка даних на мікроконтролері

Після отримання даних від сенсорів, мікроконтроллер виконує їхню обробку для отримання значень енергоспоживання.

Якщо використовуються аналогові сенсори, необхідно перетворити значення, отримані з АЦП, у фізичні величини (струм, напруга). Для цього використовуються калібрувальні коефіцієнти, які залежать від характеристик сенсора. Наприклад, для сенсора струму на основі СТ необхідно знати коефіцієнт трансформації та опір навантажувального резистора.

$$\text{Напруга на резисторі} = \frac{\text{Значення АЦП} \times V_{ref}}{1023} \quad (1.1)$$

$$\text{Струм} = \frac{\text{Напруга на резисторі}}{R_{\text{навантаження}}} \times \text{Коефіцієнт трансформації} \quad (1.2)$$

де

1. Значення АЦП – цифрове значення, отримане з аналогово-цифрового перетворювача (АЦП), яке відповідає виміряній напрузі.

2. V_{ref} – опорна напруга АЦП, максимальна напруга, яка відповідає значенню 1023 (для 10-бітного АЦП).

3. 1023 – максимальне значення для 10-бітного АЦП ($2^{10} - 1$).

4. $U_{рез}$ – напруга на вимірювальному резисторі.

5. $R_{навантаження}$ – опір навантаження, через який протікає струм.

Коефіцієнт трансформації – множник, що враховує співвідношення між вимірним і реальним струмом, якщо використовується трансформатор струму або інший перетворювач.

Для однофазної мережі миттєва активна потужність може бути розрахована як добуток миттєвих значень напруги та струму:

$$P(t) = V(t) \times I(t) \quad (1.3)$$

де

1. $P(t)$ – миттєва потужність у момент часу t , [Вт (ват)]

2. $V(t)$ – миттєве значення напруги у момент часу t , [В (вольт)]

3. $I(t)$ – миттєве значення струму у момент часу t , [А (ампер)]

Якщо напруга вважається постійною, то середню активну потужність за певний період часу можна оцінити на основі середнього значення струму:

$$P_{середня} = V_{RMS} \times I_{RMS} \quad (1.4)$$

де :

1. $P_{середня}$ – середня (активна) потужність, яка споживається або передається навантаженню, [Вт (ват)]

2. V_{RMS} – середньоквадратичне (RMS) значення напруги, [В (вольт)]. Це ефективне значення змінної напруги, яке дає той самий тепловий ефект, що й еквівалентна постійна напруга.

3. I_{RMS} – середньоквадратичне (RMS) значення струму, [А (ампер)]. Це ефективне значення змінного струму

Для більш точного вимірювання потужності при нелінійних навантаженнях необхідно вимірювати миттєві значення струму та напруги та обчислювати середнє значення їхнього добутку.

Спожита енергія за певний період часу є інтегралом потужності за цей період:

$$E = \int P(t)dt \quad (1.5)$$

де :

1. E – електрична енергія, [Дж або кВт·год]

2. $P(t)$ – миттєва потужність у часі, [Вт]

3. dt – нескінченно малий інтервал часу

Дискретно енергію можна апроксимувати як суму добутків середньої потужності за невеликі інтервали часу на тривалість цих інтервалів:

$$E \approx \sum_{i=1}^n P_{\text{середня},i} \times \Delta t_i \quad (1.6)$$

E – електрична енергія, [Дж (джоулі) або кВт·год];

$P_{\text{середня},i}$ – середня потужність на i -му інтервалі, [Вт];

Δt_i – тривалість i -го інтервалу вимірювання, [с];

n – кількість дискретних інтервалів, на які розбито загальний; час.

Якщо потужність вимірюється регулярно з інтервалом Δt , то спожиту енергію можна накопичувати, додаючи до попереднього значення добуток поточної потужності на Δt .

Для зменшення впливу шумів та випадкових коливань доцільно застосовувати фільтрацію та усереднення отриманих даних. Це може бути просте ковзне середнє або більш складні цифрові фільтри.

Після обробки отримані дані про енергоспоживання можуть бути збережені локально або передані на зовнішні пристрої чи платформи для

подальшого аналізу та візуалізації.

На багатьох типах мікроконтролерів можна організувати зберігання даних на SD-карті, підключеній через SD-кард рідер. Це корисно для автономної роботи системи та збору історичних даних. Крім того, дані можуть передаватися на комп'ютер через USB-кабель для моніторингу в реальному часі або для подальшої обробки за допомогою спеціального програмного забезпечення. За наявності Ethernet-шилда або Wi-Fi модуля, мікроконтролер може підключатися до локальної мережі або Інтернету та передавати дані на веб-сервери, хмарні платформи (наприклад, ThingSpeak, Firebase) або сервери баз даних. Це дозволяє здійснювати віддалений моніторинг та керування.

Для передачі даних можуть використовуватися різні протоколи, такі як HTTP, MQTT, TCP/IP. Вибір протоколу залежить від вимог до надійності, швидкості та складності реалізації.

Для забезпечення точності вимірювань необхідно проводити калібрування сенсорів. Це може включати порівняння показань сенсора з еталонним приладом та внесення поправок у програмний код. Регулярна перевірка та повторне калібрування можуть бути необхідними для підтримки точності протягом тривалого часу.

Точність вимірювання також залежить від якості сенсора, роздільної здатності АЦП мікроконтролера, частоти опитування та алгоритмів обробки даних. Важливо враховувати ці фактори при проектуванні системи.

Надійність збору даних є критично важливою для ефективної роботи системи контролю. Необхідно передбачити обробку можливих помилок при зчитуванні даних з сенсорів, втрати зв'язку при передачі даних та інші нештатні ситуації.

Безпека є ще одним важливим аспектом, особливо при роботі з електричними мережами високої напруги. Усі підключення повинні бути виконані кваліфікованим персоналом з дотриманням відповідних норм та правил електробезпеки. Використання неінвазивних сенсорів струму (СТ) є

більш безпечним варіантом для вимірювання струму.

Збір даних про енергоспоживання є фундаментом для побудови ефективної системи контролю на мікроконтролерах. Ретельний вибір джерел даних, відповідних сенсорів, правильне їх підключення, розробка алгоритмів обробки та забезпечення надійної передачі даних є ключовими етапами цього процесу. Врахування факторів точності, безпеки та надійності дозволить створити систему, яка надаватиме цінну інформацію для оптимізації енергоспоживання та прийняття обґрунтованих рішень. Подальша розробка системи контролю включатиме аналіз зібраних даних, візуалізацію інформації та реалізацію механізмів керування підключеними пристроями на основі отриманих значень енергоспоживання.

2 ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ КОНТРОЛЮ ТА КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

2.1 Вибір мікроконтролера

Створення ефективної системи контролю енергоспоживання на базі мікроконтролера вимагає ретельного вибору обчислювального ядра, периферії, обсягу пам'яті та інших важливих характеристик. На ринку представлено безліч мікроконтролерних платформ, кожна з яких має свої переваги та недоліки. У цьому порівнянні ми розглянемо три популярні сімейства мікроконтролерів, які часто використовуються в проектах такого типу: Arduino (на базі ATmega328P), ESP32 та STM32 (зокрема, серія STM32F1).

2.1.1 Arduino (на базі ATmega328P)

Arduino [6] (характеристики наведено у табл 2.1) є однією з найпопулярніших платформ для початківців та швидкого прототипування. Її простота, велика спільнота та величезна кількість доступних бібліотек роблять її привабливим вибором для багатьох проектів, включаючи системи контролю енергоспоживання базового рівня.

Інтегроване середовище розробки (IDE) Arduino є надзвичайно інтуїтивно зрозумілим та легким у вивченні, навіть для людей без глибоких знань у програмуванні мікроконтролерів. Його простий інтерфейс, велика кількість вбудованих прикладів коду та зрозумілі повідомлення про помилки значно полегшують початок роботи.

Arduino використовує спрощену версію C/C++, що абстрагує багато складностей, пов'язаних з безпосереднім керуванням апаратним забезпеченням мікроконтролера. Це дозволяє зосередитися на логіці системи контролю енергоспоживання, а не на низькорівневих деталях.

Для Arduino існує величезна кількість безкоштовних та добре документованих бібліотек для роботи з різноманітними датчиками (включаючи датчики струму та напруги), комунікаційними модулями (SD-карти, Ethernet Shield, ESP8266 для Wi-Fi), дисплеями та іншими периферійними пристроями. Це значно прискорює процес розробки, оскільки немає необхідності писати код з нуля для кожного компонента.

Arduino має одну з найбільших та найактивніших спільнот користувачів у світі мікроконтролерів. Це означає, що існує безліч онлайн-форумів, веб- сайтів, навчальних посібників та відеоуроків, де можна знайти відповіді на практично будь-які питання, пов'язані з розробкою проектів на Arduino. В Інтернеті доступна величезна кількість прикладів проектів, схожих на системи контролю енергоспоживання, що може слугувати відмінною відправною точкою та надати ідеї для власної розробки. У разі виникнення проблем або питань, легко отримати допомогу від інших користувачів спільноти, що значно полегшує процес налагодження та вирішення технічних труднощів.

Оригінальні плати Arduino Uno та їхні численні клони є відносно недорогими та широко доступними в багатьох електронних магазинах та онлайн-маркетплейсах. Це робить Arduino економічно вигідним вибором, особливо на етапах прототипування та для невеликих проектів. Більшість датчиків та модулів, сумісних з Arduino, також є відносно недорогими та легкодоступними.

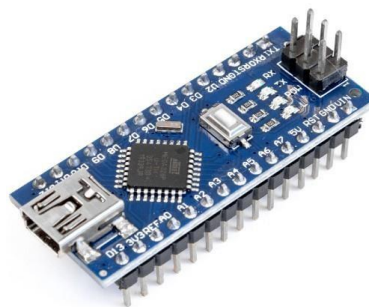


Рисунок 2.1 – Arduino Nano

Arduino Uno має 6, а Arduino Nano (відображене на рисунку 2.1) [7] - 8 аналогових входів, які можуть бути використані для підключення датчиків струму (після перетворення струму в напругу за допомогою навантажувального резистора) та датчиків напруги. 4 цифрових виходів можуть використовуватися для керування реле або твердотільними реле (SSR) для ввімкнення та вимкнення навантажень. Наявність UART, SPI та I²C дозволяє підключати різноманітні периферійні пристрої, такі як SD-кард рідери для збереження даних, Ethernet Shield або ESP8266 для підключення до мережі.

Arduino легко підключається до макетних плат та інших компонентів за допомогою з'єднувальних проводів, що дозволяє швидко збирати прототипи систем контролю енергоспоживання. Завдяки простоті IDE та великій кількості бібліотек, розробка програмного забезпечення для Arduino відбувається досить швидко, що є важливим на етапі тестування різних ідей та алгоритмів.

Важливо також усвідомлювати обмеження Arduino. Для складних алгоритмів обробки даних у реальному часі, великих обсягів збережених даних або реалізації розширених функцій (наприклад, машинне навчання) ресурси Arduino можуть бути недостатніми.

Для проектів, що потребують безпосереднього підключення до мережі без використання зовнішніх модулів, ESP32 може бути кращим вибором. Для застосувань, що вимагають високої точності вимірювань енергоспоживання, 12-бітний АЦП, наявний в ESP32 або STM32, може бути більш прийнятним.

Незважаючи на певні обмеження, Arduino є чудовим вибором для створення систем контролю енергоспоживання, особливо на початкових етапах розробки, для освітніх цілей, а також для проектів з невеликою або середньою складністю, де не потрібна інтенсивна обробка даних або вбудоване підключення до мережі. Його простота використання, велика спільнота, низька вартість та достатня функціональність для багатьох

базових завдань роблять його привабливою платформою для швидкого прототипування та реалізації функціональних систем моніторингу та керування енергоспоживанням. У випадках, коли потрібна більша обчислювальна потужність, більше пам'яті, вбудоване підключення до мережі або вища точність аналогового введення, слід розглянути альтернативні мікроконтролерні платформи, такі як ESP32 або STM32. Однак, для багатьох початкових та середніх проектів Arduino залишається оптимальним рішенням.

2.1.2 ESP32

Однією з ключових переваг ESP32 (відображене на рисунку 2.2) є вбудована підтримка Wi-Fi (802.11 b/g/n). Це дозволяє легко інтегрувати систему контролю енергоспоживання в локальну мережу або безпосередньо підключати її до Інтернету без використання додаткових дорогих та складних Ethernet Shield або окремих Wi-Fi модулів. Це спрощує передачу даних на хмарні платформи (наприклад, ThingSpeak, Firebase), веб-сервери для віддаленого моніторингу та керування, а також реалізацію бездротових оновлень програмного забезпечення (OTA). Наявність вбудованого Bluetooth та BLE відкриває додаткові можливості для локального зв'язку з іншими пристроями (наприклад, смартфонами для налаштування та моніторингу), створення mesh-мереж для розподілених систем збору даних або використання BLE для низькоенергетичної передачі даних з датчиків.

ESP32 зазвичай оснащується потужним 32-бітним двоядерним мікропроцесором Tensilica LX6 з тактовою частотою до 240 МГц. Це забезпечує значно більшу обчислювальну потужність порівняно з 8-бітними мікроконтролерами, що дозволяє виконувати складні алгоритми обробки даних у реальному часі, такі як просунута фільтрація сигналів, точний розрахунок діючої потужності для несинусоїдальних навантажень, швидкий збір даних з багатьох датчиків та навіть базові алгоритми машинного

навчання для інтелектуального керування енергоспоживанням. ESP32 має значний обсяг флеш-пам'яті (зазвичай 4 МБ або більше) та SRAM (до 520 КБ), що дозволяє зберігати великі обсяги історичних даних про енергоспоживання, використовувати складніші програмні бібліотеки та реалізовувати розширені функціональні можливості.

ESP32 надає велику кількість аналогових входів з 12-бітною роздільною здатністю (4096 рівнів), що забезпечує точніші вимірювання аналогових сигналів від датчиків струму та напруги. Більша кількість каналів дозволяє підключати більше датчиків для моніторингу різних точок енергоспоживання без необхідності використання мультиплексорів.



Рисунок 2.2 - ESP32

ESP32 має кілька апаратних інтерфейсів UART, SPI та I²C, що забезпечує гнучкість при підключенні різноманітних периферійних пристроїв, включаючи датчики, дисплеї, SD-кард рідери та інші комунікаційні модулі. Наявність I²S (для обробки аудіо), CAN 2.0 (для промислових застосувань) та Ethernet (в деяких моделях) розширює можливості інтеграції ESP32 в різноманітні системи контролю енергоспоживання.

ESP32 має розвинені режими глибокого сну з дуже низьким енергоспоживанням, що є критично важливим для систем, які можуть

працювати від батарей або потребують мінімізації власного енергоспоживання. Мікроконтролер може прокидатися за таймером, зовнішнім перериванням або іншими подіями для збору даних або керування навантаженнями, а потім знову переходити в режим сну для економії енергії.

ESP32 має велику та активну спільноту розробників, що забезпечує доступ до численних прикладів коду, бібліотек, навчальних посібників та форумів для вирішення проблем. Виробник ESP32, компанія Espressif Systems, надає детальну та якісну документацію на свої мікроконтролери та SDK (Software Development Kit), що полегшує процес розробки. ESP32 добре підтримується в популярному середовищі розробки Arduino IDE (через спеціальні розширення), а також має власну потужну платформу розробки ESP-IDF, що надає більший контроль над апаратним забезпеченням.

Проста інтеграція з популярними сервісами: Вбудований Wi-Fi робить ESP32 ідеальним вибором для підключення до різноманітних хмарних платформ Інтернету речей (IoT), таких як AWS IoT Core, Google Cloud IoT Platform, Microsoft Azure IoT Hub, ThingSpeak, Firebase та інших. Існують готові бібліотеки та приклади для спрощення цього процесу, що дозволяє легко передавати дані про енергоспоживання для зберігання, візуалізації та аналізу в хмарі.

Однак мікроконтролери ESP32 мають вищу вартість порівняно з Arduino. Плати на базі ESP32 зазвичай є дещо дорожчими, ніж Arduino Uno. При активному використанні Wi-Fi ESP32 може споживати більше енергії, ніж Arduino. Хоча підтримка в Arduino IDE спрощує початок роботи, більша кількість функцій та можливостей ESP32 може здаватися дещо складнішою для тих, хто тільки починає знайомство з мікроконтролерами.

Вибір ESP32 для системи контролю енергоспоживання може бути обґрунтованим завдяки його вбудованому Wi-Fi та Bluetooth для бездротового зв'язку, потужним обчислювальним ресурсам для складної обробки даних, достатній кількості аналогових входів з високою роздільною здатністю для точних вимірювань, широкому спектру периферійних

інтерфейсів для підключення різноманітних датчиків та модулів, а також можливості роботи з низьким енергоспоживанням. Однак, якщо проект не вимагає складної системи контролю, масштабування та зв'язку з хмарними ресурсами, застосування ESP32 може бути зайвим.

2.1.3 STM32

Мікроконтролери STM32 (відображене на рисунку 2.3) засновані на потужних 32-бітних ядрах ARM Cortex-M (різних серій, наприклад, M0, M3, M4, M7), які забезпечують значно вищу обчислювальну потужність порівняно з 8-бітними ядрами, що використовуються в Arduino. Це дозволяє реалізовувати складніші алгоритми обробки даних у реальному часі, такі як просунуті методи фільтрації сигналу, розрахунок діючої потужності (RMS) для несинусоїдальних сигналів, швидкий збір даних з багатьох датчиків та виконання інших ресурсоемних завдань.

STM32 мікроконтролери зазвичай мають вищу тактову частоту (наприклад, 72 МГц і вище для серії F1, значно вище для більш сучасних серій), що також сприяє швидшій обробці даних та кращій реакції системи.

Багато мікроконтролерів STM32 оснащені 12-бітними або навіть вищими розрядностями аналогово-цифровими перетворювачами (АЦП) з більшою кількістю каналів порівняно з Arduino (10-бітний АЦП). Це забезпечує значно вищу роздільну здатність при вимірюванні аналогових сигналів від датчиків струму та напруги, що призводить до точніших даних про енергоспоживання. Вища точність особливо важлива для виявлення невеликих змін у споживанні та для точного обліку енергії.

STM32 мікроконтролери зазвичай мають більшу кількість апаратних інтерфейсів UART, SPI та I²C, що забезпечує гнучкість при підключенні різноманітних периферійних пристроїв, таких як датчики, дисплеї, модулі зв'язку (наприклад, для SD-карт, Ethernet, Wi-Fi через SPI/UART). Деякі серії STM32 мають вбудовані інтерфейси CAN (для промислових мереж), USB

(для зв'язку з комп'ютером або іншими пристроями) та Ethernet (для підключення до локальної мережі), що може бути корисним для більш складних систем контролю енергоспоживання з розширеними можливостями зв'язку.



Рисунок 2.3 - STM32 F4DISCOVERY

Сімейство STM32 включає велику кількість серій та моделей мікроконтролерів з різними характеристиками за ціною, продуктивністю, обсягом пам'яті та набором периферії. Це дозволяє вибрати оптимальний мікроконтролер для конкретних вимог системи контролю енергоспоживання, починаючи від простих завдань і закінчуючи складними промисловими рішеннями.

Багато мікроконтролерів STM32 підтримують широкий промисловий температурний діапазон, що робить їх придатними для використання в суворих умовах експлуатації. STMicroelectronics зазвичай забезпечує тривалий термін підтримки своїх мікроконтролерів, що є важливим для довгострокових проектів.

Головні характеристики мікроконтролерів наведені в таблиці 2.1.

Професійні IDE та інструменти розробки: Для STM32 доступні потужні та професійні інтегровані середовища розробки (IDE), такі як STM32CubeIDE (безкоштовний від STMicroelectronics), Keil MDK та IAR Embedded Workbench, які надають широкі можливості для налагодження, профілювання та оптимізації коду. В останній час можливість програмування

STM32 доступно і в Arduino IDE. STMicroelectronics надає рівні абстракції апаратного забезпечення (HAL) та низькорівневі (LL) API, що дозволяє розробникам вибирати між простотою використання та гнучкістю керування апаратним забезпеченням.

Таблиця 2.1 – Головні характеристики мікроконтролерів.

Характеристика	Arduino Nano (ATmega328P)	ESP32	STM32F1 (STM32F103C8T6)
1	2	3	4
Ядро	8-бітне AVR	32-бітне Tensilica LX6	32-бітне ARM Cortex-M3
Тактова частота	16 МГц	до 240 МГц	72 МГц
Флеш-пам'ять	32 КБ	≥ 4 МБ	64/128 КБ
SRAM	2 КБ	до 520 КБ	20 КБ
АЦП	6 x 10-біт	до 18 x 12-біт	до 16 x 12-біт
Wi-Fi	Ні	Вбудований	Ні
Bluetooth	Ні	Вбудований	Ні
Ethernet	Зовнішній модуль	Вбудований (в деяких моделях)	Зовнішній модуль
Кількість GPIO	22	до 34	до 37
Інтерфейси	UART, SPI, I ² C	UART, SPI, I ² C, I ² S, CAN	UART, SPI, I ² C, CAN, USB
Простота використання	Дуже висока	Середня	Середня - низька
Вартість плати	Низька	Середня	Низька (для базових плат)
Енергоспоживання	Низьке	Середнє - високе (активний режим Wi-Fi)	Низьке

Потужний графічний інструмент STM32CubeMX для конфігурації мікроконтролера, налаштування периферії, генерації коду ініціалізації та керування живленням значно спрощує процес розробки.

Однак, робота з STM32 може бути складнішою для новачків порівняно з Arduino через більшу кількість функцій, складнішу архітектуру та різноманітність інструментів розробки.

Хоча спільнота STM32 є досить великою, кількість готових бібліотек для Arduino може бути більшою, особливо для дуже специфічних або нішевих датчиків.

Вибір STM32 для системи контролю енергоспоживання є обґрунтованим, коли проект вимагає високої продуктивності для складної обробки даних, точних вимірювань енергоспоживання завдяки АЦП високої роздільної здатності, гнучкості в підключенні великої кількості різноманітних периферійних пристроїв через численні апаратні інтерфейси, а також надійності та можливості застосування в промислових умовах. Хоча початковий поріг входу для розробки на STM32 може бути вищим, потужні інструменти розробки та широкі можливості мікроконтролерів цієї родини роблять їх відмінним вибором для більш вимогливих та складних систем контролю енергоспоживання. Для простих прототипів або проектів з обмеженим бюджетом та часом розробки Arduino може залишатися більш прийнятним варіантом.

Остаточний вибір Arduino для проектуємої системи контролю енергоспоживання обґрунтовується передусім простотою розробки та великою кількістю готових рішень, що є критично важливим для швидкого прототипування та реалізації базової функціональності. Величезна спільнота користувачів та доступність безлічі бібліотек для роботи з датчиками струму, напруги та комунікаційними модулями значно скорочують час розробки та полегшують вирішення потенційних проблем. Крім того, низька вартість плати Arduino та сумісних компонентів робить її економічно привабливим

варіантом, особливо на етапі експериментів та тестування різних підходів до моніторингу та керування енергоспоживанням. Легкість підключення до макетних плат та інших компонентів спрощує процес створення прототипу, дозволяючи швидко перевірити працездатність основних функцій системи.

Таким чином, вибір Arduino для даної системи контролю енергоспоживання обґрунтовується балансом між простотою використання, вартістю, доступністю ресурсів та достатньою функціональністю для реалізації базових та середніх за складністю завдань моніторингу та керування. Велика спільнота та наявність готових рішень значно прискорюють процес розробки та роблять Arduino надійним та перевіреним вибором для мого проекту.

2.2 Вибір сенсорів струму для системи контролю енергоспоживання

Ефективний моніторинг та керування енергоспоживанням є ключовим аспектом сучасних енергетичних систем, будь то розумний будинок, промислове підприємство чи система альтернативної енергетики. Центральну роль у таких системах відіграють датчики струму, які забезпечують необхідні дані про величину електричного струму, що протікає через різні елементи системи. Правильний вибір датчика струму для використання з Arduino є критично важливим для забезпечення точності вимірювань, безпеки експлуатації та відповідності вимогам конкретного застосування.

На ринку представлений широкий спектр датчиків струму, які відрізняються за принципом дії, характеристиками, точністю, вартістю та зручністю інтеграції з Arduino. У цьому описі ми детально розглянемо основні типи датчиків струму, придатних для використання в системах контролю енергоспоживання на базі Arduino, порівняємо їхні ключові особливості, переваги та недоліки, а також надамо рекомендації щодо вибору оптимального датчика для різних сценаріїв застосування.

2.2.1 Сенсор струму ACS712

Сенсор струму ACS712 (відображене на рисунку 2.4) є популярним вибором для систем контролю енергоспоживання на базі Arduino, особливо в проектах початкового та середнього рівня складності, де важливими факторами є простота використання, можливість вимірювання як постійного (DC), так і змінного (AC) струму, а також відносно невисока вартість. Його інтегрована структура та зручний інтерфейс роблять його привабливим варіантом для широкого кола застосувань.



Рисунок 2.4 – Сенсор струму ACS712

Однією з ключових переваг ACS712 є його здатність вимірювати як постійний (DC), так і змінний (AC) струм. Це робить його універсальним рішенням для систем контролю енергоспоживання, які можуть включати різні типи навантажень та джерел енергії. Наприклад, він може використовуватися для моніторингу споживання енергії побутовими приладами, які працюють від мережі змінного струму, а також для вимірювання струму в акумуляторних системах або від сонячних панелей (до інвертора), де використовується постійний струм. Ця універсальність спрощує проектування системи, оскільки один тип сенсора може використовуватися для різних етапів моніторингу.

ACS712 є інтегрованою мікросхемою, яка об'єднує датчик струму на

основі ефекту Холла та схему обробки сигналу. Це значно спрощує його використання порівняно з окремими трансформаторами струму, які потребують додаткового навантажувального резистора та схеми зсуву напруги для вимірювання змінного струму за допомогою Arduino. ACS712 зазвичай має простий аналоговий вихід, напруга якого лінійно залежить від вимірюваного струму. Це дозволяє безпосередньо підключати вихід сенсора до аналогового входу Arduino та зчитувати значення струму за допомогою функції `analogRead()`.

ACS712 доступний у компактному корпусі SOIC-8, що полегшує його інтеграцію в різні електронні пристрої та макетні плати. Існують також готові модулі на основі ACS712 з зручними роз'ємами (наприклад, гвинтовими клемми або піновими роз'ємами), які спрощують підключення до Arduino та силових ланцюгів. Невеликі розміри сенсора роблять його придатним для застосувань з обмеженим простором.

Хоча ACS712 забезпечує певну ступінь гальванічної ізоляції між вимірюваним струмом та вихідним сигналом (зазвичай до кількох кіловольт), слід зазначити, що ця ізоляція є обмеженою і не призначена для застосувань з високою напругою, де потрібна посилена ізоляція для безпеки. Для високовольтних систем перевагу слід надавати трансформаторам струму з повною гальванічною ізоляцією. Однак для низьковольтних систем, таких як моніторинг споживання окремих побутових приладів або струму в акумуляторних ланцюгах, рівень ізоляції ACS712 може бути достатнім.

ACS712 є досить доступним за ціною сенсором струму, особливо в порівнянні з деякими більш точними або спеціалізованими датчиками на основі ефекту Холла або котушок Роговського. Його низька вартість робить його привабливим вибором для бюджетних проектів та для використання у великій кількості точок моніторингу в системі.

Вихідна напруга ACS712 лінійно залежить від вимірюваного струму, що спрощує калібрування та перетворення аналогових значень, отриманих з Arduino, у значення струму. У технічній документації (datasheet) на сенсор

зазвичай наводяться коефіцієнти перетворення (mV/A) для різних моделей (наприклад, 5A, 20A, 30A версії).

ACS712 доступний у різних версіях з обмеженими діапазонами вимірювання струму (наприклад, $\pm 5A$, $\pm 20A$, $\pm 30A$). Він не підходить для вимірювання дуже великих струмів, які можуть зустрічатися в промислових установках або на головних вводах електромережі.

Вихідний аналоговий сигнал може бути чутливим до електричних шумів, тому може знадобитися фільтрація сигналу за допомогою апаратних (RC-фільтр) або програмних методів на Arduino.

Хоча теоретично існують способи неінвазивного застосування ACS712, найчастіше він використовується з розривом електричного кола та послідовним включенням струмопровідника через спеціальні отвори в корпусі мікросхеми або на модулі. Це може бути незручним або небезпечним у деяких ситуаціях.

Використання сенсора струму ACS712 для системи контролю енергоспоживання на Arduino є обґрунтованим вибором для проектів, де важливими є простота використання, можливість вимірювання як постійного, так і змінного струму в обмеженому діапазоні, компактність, відносно низька вартість та наявність лінійного аналогового виходу. Він особливо добре підходить для моніторингу споживання окремих побутових приладів, струму в низьковольтних акумуляторних системах та інших подібних застосувань, де не потрібні вимірювання дуже великих струмів або висока точність. Однак, при роботі з високовольтними системами або при необхідності вимірювання великих струмів слід розглянути альтернативні датчики, такі як трансформатори струму або спеціалізовані датчики Холла з відповідними характеристиками безпеки та діапазоном вимірювання. Враховуючи його простоту інтеграції та універсальність, ACS712 є хорошим варіантом для багатьох розробників, які створюють системи контролю енергоспоживання на базі Arduino.

2.2.2 Сенсор струму SCT-013-000

Сенсор струму SCT-013-000 (відображене на рисунку 2.5) є надзвичайно популярним вибором для систем контролю енергоспоживання на базі Arduino, особливо коли йдеться про моніторинг змінного струму (AC) в побутових та комерційних електромережах. Його неінвазивна конструкція, широкий діапазон вимірювання, відносно низька вартість та простота інтеграції з Arduino роблять його одним з найбільш привабливих варіантів для багатьох застосувань.



Рисунок 2.5 – Сенсор струму SCT-013-000

Однією з ключових переваг SCT-013-000 є його неінвазивна конструкція з роз'ємним сердечником (split-core). Це означає, що для встановлення датчика не потрібно розривати електричне коло. Датчик просто заціпається навколо провідника, струм якого необхідно виміряти. Це значно спрощує процес монтажу, особливо в існуючих електричних установках, та підвищує безпеку користувача, оскільки немає прямого контакту з провідниками під напругою. Ця особливість робить SCT-013-000 ідеальним для моніторингу споживання енергії без втручання в існуючу проводку.

SCT-013-000 спеціально розроблений для вимірювання змінного струму, який є основним типом струму, що використовується в більшості побутових та комерційних електромереж. Його діапазон вимірювання струму

зазвичай становить до 100 Ампер, що охоплює споживання більшості побутових приладів, освітлення, а також окремих ліній живлення в будинку чи офісі. Цей широкий діапазон робить його придатним для моніторингу як невеликих навантажень, так і значного споживання енергії.

Трансформаторна природа SCT-013-000 забезпечує гальванічну ізоляцію між вимірюваним колом високої напруги та низьковольтним колом Arduino. Це є критично важливим для безпеки, оскільки запобігає потраплянню високої напруги на мікроконтролер та захищає користувача від ураження електричним струмом. Ця ізоляція є значною перевагою порівняно з інвазивними датчиками, такими як резистивні шунти, які не забезпечують гальванічної розв'язки.

SCT-013-000 є одним з найбільш доступних за ціною неінвазивних датчиків струму на ринку. Його популярність та широке використання призвели до його доступності у багатьох електронних магазинах та онлайн-маркетплейсах. Низька вартість робить його економічно вигідним рішенням для проектів, де потрібно контролювати споживання енергії в багатьох точках або для великомасштабних систем моніторингу.

Для використання SCT-013-000 з Arduino потрібен лише навантажувальний резистор (burden resistor), який перетворює вихідний струм датчика у пропорційну напругу, яку може зчитати аналоговий вхід Arduino. Значення цього резистора вибирається залежно від бажаного діапазону вихідної напруги та максимального вимірюваного струму. Існують численні приклади коду та навчальні матеріали, які детально описують процес підключення та калібрування SCT-013-000 для використання з Arduino.

Оскільки більшість побутових та комерційних приладів працюють від мережі змінного струму, SCT-013-000 є ідеальним вибором для моніторингу їхнього енергоспоживання. Він може використовуватися для вимірювання споживання окремих приладів (холодильників, пральних машин, кондиціонерів), груп приладів (освітлення, розетки в кімнаті) або навіть

загального споживання окремих ліній електропроводки.

Важливо також враховувати певні обмеження SCT-013-000, він вимірює лише змінний струм (AC): SCT-013-000 не підходить для вимірювання постійного струму (DC), що обмежує його застосування в системах з акумуляторами або сонячними панелями без інвертора. Для отримання точних значень струму необхідно правильно підібрати навантажувальний резистор та виконати калібрування системи, враховуючи характеристики датчика та АЦП Arduino. Точність вимірювання може бути не оптимальною при дуже малих струмах, оскільки вихідний сигнал також буде невеликим.

Використання сенсора струму SCT-013-000 для системи контролю енергоспоживання на Arduino є обґрунтованим вибором завдяки його неінвазивній та безпечній конструкції, здатності вимірювати змінний струм у широкому діапазоні, гальванічній ізоляції, відносно низькій вартості та простоті інтеграції з Arduino. Він є ідеальним рішенням для моніторингу енергоспоживання більшості побутових та комерційних електричних приладів та ліній живлення. Хоча він не підходить для вимірювання постійного струму, його переваги роблять його одним з найбільш популярних та ефективних датчиків для багатьох завдань контролю енергоспоживання на базі Arduino.

2.2.3 Сенсора струму та напруги INA219

Сенсор INA219 (відображене на рисунку 2.6) є високоточною та універсальною інтегральною мікросхемою, яка забезпечує вимірювання не лише струму, але й напруги та потужності в DC-ланцюгах. Його зручний інтерфейс I²C та інтегрована схема роблять його чудовим вибором для систем контролю енергоспоживання на базі Arduino, особливо в застосуваннях з постійним струмом, таких як моніторинг сонячних панелей, акумуляторних систем, живлення DC-двигунів та інших DC-навантажень.

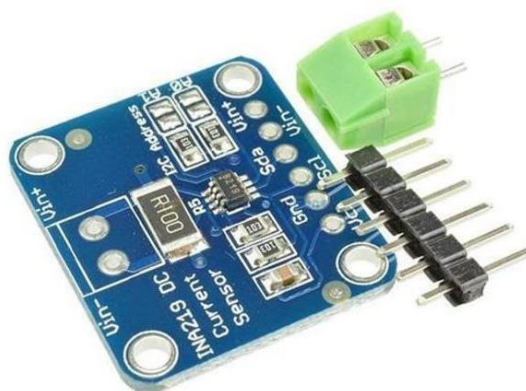


Рисунок 2.6 – Сенсор струму INA219

Основною перевагою INA219 є його здатність одночасно вимірювати струм, напругу (на шунті та на навантаженні) та обчислювати потужність. Це надає повну картину енергоспоживання в DC-ланцюзі без необхідності використання кількох окремих сенсорів та складних розрахунків на Arduino. Отримання даних про напругу дозволяє не лише контролювати струм, але й відстежувати стан джерела живлення або навантаження, що є важливим для діагностики та оптимізації енергоефективності.

INA219 є високоточним сенсором, що забезпечує точні вимірювання струму та напруги завдяки своєму 12-бітному АЦП. Це особливо важливо для систем контролю енергоспоживання, де потрібен точний облік спожитої або виробленої енергії для аналізу ефективності, виявлення аномалій або реалізації алгоритмів керування навантаженням. Висока точність дозволяє виявляти навіть невеликі зміни в енергоспоживанні.

INA219 використовує простий та поширений цифровий інтерфейс I²C (також відомий як TWI) для зв'язку з Arduino. Це означає, що для отримання даних потрібно лише два дроти (SDA та SCL) плюс живлення та земля. I²C є синхронним протоколом, який дозволяє підключати кілька пристроїв до однієї шини, що є зручним для розширення системи моніторингу з використанням кількох сенсорів або інших I²C-сумісних пристроїв. Цифрова

передача даних також зменшує вплив шумів на довгих лініях зв'язку порівняно з аналоговими сигналами.

Багато готових модулів на основі INA219 вже мають вбудований прецизійний шунтовий резистор з низьким опором (зазвичай 0.1 Ом). Це спрощує підключення сенсора до силового ланцюга, оскільки користувачеві не потрібно самостійно підбирати та встановлювати шунт. Використання прецизійного шунта забезпечує точні вимірювання струму.

INA219 здатний вимірювати струм в обох напрямках, що є важливим для систем з рекуперацією енергії або двонаправленим потоком енергії, таких як системи з акумуляторами, які можуть як заряджатися, так і розряджатися. Це робить його придатним для моніторингу складних енергетичних систем.

INA219 має програмовані параметри, такі як діапазон вимірювання струму та напруги, а також можливість калібрування для підвищення точності вимірювань. Це дозволяє налаштувати сенсор під конкретні вимоги системи та отримати більш достовірні дані. INA219 має низьке власне енергоспоживання, що є важливим для систем, які працюють від батарей або де необхідно мінімізувати загальне енергоспоживання системи контролю.

Однак, INA219 призначений для вимірювання струму та напруги в DC-ланцюгах і не підходить для безпосереднього вимірювання змінного струму (AC) без додаткових схем перетворення. Для вимірювання струму INA219 необхідно включати послідовно в електричне коло через шунтовий резистор. Це вимагає розриву кола, що може бути незручним або небезпечним у деяких ситуаціях. Діапазон вимірюваного струму залежить від опору шунта. Для вимірювання великих струмів потрібні шунти з дуже низьким опором, що може призвести до меншої роздільної здатності вимірювання напруги на шунті.

Використання сенсора струму та напруги INA219 є обґрунтованим вибором для систем контролю енергоспоживання на Arduino, які працюють з постійним струмом, де потрібне комплексне вимірювання струму, напруги та

потужності з високою точністю та зручним цифровим інтерфейсом I²C. Його інтегрована структура, можливість вимірювання двонаправленого струму та програмовані параметри роблять його потужним інструментом для моніторингу та аналізу енергоспоживання в різноманітних DC-застосуваннях, таких як системи на сонячних батареях, акумуляторні системи, керування DC-двигунами та інші. Однак, для моніторингу змінного струму (AC) слід використовувати інші типи сенсорів, такі як трансформатори струму (SCT-013) або інтегровані сенсори ACS712.

Для порівняння вартість сенсорів ACS712 в Україні складає приблизно

60 грн, сенсорів SCT-013 – приблизно 280 грн, а сенсорів INA219 – приблизно 100 грн.

Тому враховуючі аналіз можливих рішень, для нашого навчального проекту найкращім варіантом буде застосування найбільш універсального сенсору типу ACS712 на максимальний струм 5А, щоб забезпечити максимальну точність вимірювання електроспоживання низької інтенсивності.

2.3 Вибір засобу відображення інформації

Вибір засобу відображення інформації для системи контролю енергоспоживання на Arduino залежить від кількох факторів, включаючи складність інформації, необхідний рівень деталізації, наявні ресурси Arduino, бюджет проекту, зручність використання та естетичні вимоги.

2.3.1 Семисегментні індикатори

Відображають цифри та деякі літери за допомогою комбінації семи світлодіодних сегментів. Як головна властивість – простота підключення та програмування. Для відображення простих числових значень (наприклад, поточне споживання у ватах або амперах) потрібна невелика кількість пінів Arduino та прості бібліотеки. Крім того, семисегментні індикатори є одними з

найдешевших засобів відображення. Забезпечують чітке відображення цифр. (Семисегментний індикатор відображений на рисунку 2.7)



Рисунок 2.7 – Семисегментний індикатор

Однак, мають певні недоліки. Можуть відображати лише цифри та деякі літери, що робить їх непридатними для відображення складних текстових повідомлень або графіків. Для відображення багатозначних чисел потрібно використовувати кілька семисегментних індикаторів, що збільшує кількість зайнятих пінів Arduino або потребує використання зсувних регістрів.

2.3.2 Рідкокристалічні дисплеї (LCD) на основі HD44780

Відображають символи на матриці з рідких кристалів. Найпоширеніші варіанти мають 16 символів у 2 рядки (16x2) або 20 символів у 4 рядки (20x4). Можуть відображати як числові значення, так і короткі текстові повідомлення про стан системи, одиниці вимірювання тощо.

Відносно просте підключення та програмування. Існують зручні бібліотеки для Arduino, які полегшують керування LCD-дисплеями. Можливе підключення як безпосередньо (використовуючи багато пінів), так і через I²C-модуль (використовуючи лише 2 піни даних). LCD-дисплеї на основі HD44780 є досить доступними. Сучасна вартість в Україні дисплея LCD1602 з I²C-модулем складає приблизно 100 грн. (Дисплей LCD1602 з I²C-модулем відображений на рисунку 2.8)



Рисунок 2.8 – Дисплей LCD1602 з I²C-модулем

До недоліків можна віднести – обмежені графічні можливості. Не підходять для відображення складних графіків або візуалізацій. Потрібне підсвічування для видимості в темряві. Без підсвічування інформація на дисплеї може бути погано видно. Споживають більше енергії, ніж семисегментні індикатори (особливо з підсвічуванням).

2.3.3 OLED-дисплеї

Використовують органічні світлодіоди, які світяться при проходженні через них електричного струму. Кожен піксель є самостійним джерелом світла. Доступні в компактних розмірах, що підходить для портативних або вбудованих систем. Висока контрастність та яскравість. Забезпечують чітке та добре видиме зображення навіть без підсвічування. Низьке енергоспоживання (для відображення невеликої кількості яскравих пікселів).

Можливість відображення графіки та текст. Деякі OLED-дисплеї (відображений на рисунку 2.9) (наприклад, на контролері SSD1306) підтримують відображення графічних елементів, діаграм та шрифтів різних розмірів. Зазвичай використовують I²C або SPI, що потребує невеликої кількості пінів Arduino.



Рисунок 2.9 – OLED-дисплей

До недоліків можна віднести - OLED-дисплеї зазвичай дорожчі за LCD- дисплеї на основі HD44780. Чутливість до тривалого відображення статичних зображень ("вигорання" пікселів). Можуть споживати більше енергії при відображенні великої кількості яскравих пікселів.

2.3.4 Графічні LCD-дисплеї (GLCD)

Рідкокристалічні дисплеї з матрицею пікселів, що дозволяє відображати складну графіку та текст різних розмірів. Ідеально підходять для візуалізації даних про енергоспоживання у вигляді графіків (наприклад, споживання за певний період часу). Дозволяють створювати користувацькі інтерфейси з кнопками, індикаторами тощо.

Однак керування GLCD (відображений на рисунку 2.10) вимагає більш складного коду та використання спеціальних бібліотек (наприклад, U8glib, Adafruit GFX). Потрібна більша кількість пінів Arduino (або використання послідовних інтерфейсів). Вища вартість порівняно з символічними LCD та семисегментними індикаторами. Можуть споживати більше енергії.

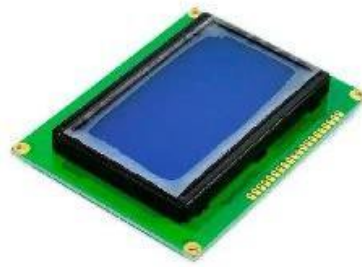


Рисунок 2.10 – Графічний LCD-дисплей

2.3.5 TFT LCD-дисплеї

Тонкоплівкові транзисторні рідкокристалічні дисплеї (відображений на рисунку 2.11), які забезпечують кольорове відображення та високу роздільну здатність. Забезпечують можливість візуалізації даних у кольорі, що може зробити інтерфейс більш інформативним та привабливим. Дозволяють відображати складну графіку, детальні діаграми та велику кількість тексту. Можливість інтеграції тачскріна. Забезпечують зручний спосіб керування системою без використання фізичних кнопок.



Рисунок 2.11 TFT LCD-дисплей

Однак TFT LCD-дисплеї є найдорожчим варіантом серед перелічених. Мають складне підключення та програмування. Керування TFT-дисплеями

вимагає значних обчислювальних ресурсів Arduino та використання складних бібліотек. Значне споживання енергії. Особливо дисплеї великого розміру та з високою яскравістю можуть споживати значну кількість енергії. Потрібна велика кількість пінів Arduino (або використання спеціальних інтерфейсів).

2.3.6 Передача даних на комп'ютер або смартфон

Як варіант – відображення інформації не на фізичному дисплеї, підключеному до Arduino, а на зовнішньому пристрої (комп'ютері через послідовний порт або смартфоні через Bluetooth/Wi-Fi).

При цьому - не вимагаються додаткові витрати на дисплей. Використовується вже існуючий пристрій. Можливість відображення великого обсягу інформації та складних візуалізацій. Обмеження на розмір екрана та обчислювальні ресурси значно менші, ніж у вбудованих дисплеїв Arduino. Можливість створення графічних інтерфейсів користувача (GUI) на комп'ютері або в мобільному додатку.

Однак при цьому вимагається постійне підключення до комп'ютера або наявність бездротового зв'язку. Система не є повністю автономною у плані відображення інформації.

Може бути менш зручним для постійного моніторингу. Потрібно мати доступ до комп'ютера або смартфона.

Вибір конкретного засобу відображення залежить від наших потреб:

- Для простого відображення поточного споживання – семисегментні індикатори або невеликий LCD (16x2) можуть бути достатніми.
- Для відображення детальної інформації та текстових повідомлень – LCD (16x2 або 20x4) або невеликий OLED-дисплей.
- Для візуалізації графіків та діаграм енергоспоживання – графічний LCD (128x64) або OLED-дисплей з підтримкою графіки.
- Для створення інтерактивного інтерфейсу з кольоровим

відображенням – TFT LCD-дисплей з тачскріном.

- Для відображення великого обсягу даних: Передача даних на комп'ютер або смартфон.

Тому, що я створюю навчальний проект, то зупинюсь на простому та недорогому варіанті, такому як LCD 16x2 (з I²C-модулем для спрощення підключення до мікроконтролеру). Якщо, в подальшому, виникне необхідність, то можна буде перейти до більш складних та інформативних засобів відображення.

2.4 Вибір засобів керування енергоспоживанням

Вибір засобів керування енергоспоживанням для системи на Arduino залежить від типу навантажень, необхідного рівня контролю (ввімкнення/вимкнення, плавне регулювання), потужності керованих пристроїв, вимог до безпеки та бюджету проекту.

2.4.1 Реле (Electromechanical Relays)

Електромеханічний перемикач, який використовує електромагніт для механічного замикання або розмикання контактів, через які протікає струм навантаження. Здатен комутувати як АС, так і DC навантаження. Універсальні для більшості типів електричних пристроїв. Висока потужність комутації. Доступні реле, здатні керувати значними струмами та напругами. Забезпечують повну електричну ізоляцію між низьковольтним колом керування Arduino та високовольтним колом навантаження, що є важливим для безпеки. Реле є відносно недорогими компонентами.

З іншого боку мають обмежений термін служби, що визначається кількістю циклів ввімкнення/вимкнення. Механічне замикання/розмикання контактів супроводжується клацанням. Механічна інерція обмежує максимальну частоту перемикання. Для керування реле від Arduino потрібен

транзисторний ключ (драйвер), оскільки реле споживають більший струм, ніж можуть безпосередньо забезпечити виходи Arduino.

2.4.2 Твердотільні реле (Solid State Relays - SSR)

Електронні перемикачі, які використовують напівпровідникові елементи (симістори, тріаки, MOSFET) для комутації навантаження. Керування здійснюється за допомогою низьковольтного сигналу, який оптично ізольований від силового кола. Не мають механічних рухомих частин, тому працюють безшумно. Можуть комутувати навантаження з високою частотою. Не мають механічного зносу, що забезпечує значно довший термін служби порівняно з електромеханічними реле. Забезпечують оптичну ізоляцію між колом керування та колом навантаження. Відсутність механічних контактів робить їх більш стійкими до зовнішніх впливів.

Однак, SSR зазвичай дорожчі за електромеханічні реле з аналогічними характеристиками. На відкритому SSR може спостерігатися невелике падіння напруги, що призводить до невеликих втрат потужності та нагрівання. При комутації значних струмів SSR можуть сильно нагріватися і потребувати радіаторів для відведення тепла. Існують SSR для AC та DC навантажень, тому потрібно вибирати відповідний тип.

2.4.3 Симістори та тріаки

Напівпровідникові прилади, які використовуються для керування потужністю в AC-ланцюгах шляхом фазового регулювання (зміни моменту ввімкнення синусоїди). Дозволяють не лише вмикати/вимикати AC-навантаження, але й плавно регулювати їхню потужність (наприклад, яскравість ламп, швидкість обертання двигунів). Симістори та тріаки можуть керувати значними струмами та напругами. Відносно невисока вартість (окремих компонентів).

Однак, призначені виключно для керування змінним струмом. Для

реалізації фазового регулювання потрібні більш складні схеми керування та алгоритми на Arduino, включаючи синхронізацію з нулем синусоїди. Для безпечного керування симісторами/тріаками від Arduino потрібна оптоізоляція. Можуть створювати електромагнітні завади (ЕМІ).

2.4.4 Транзистори (MOSFET, BJT)

Напівпровідникові прилади, які можуть використовуватися як електронні ключі для керування DC-навантаженнями. MOSFETи зазвичай кращі для керування потужними навантаженнями завдяки нижчому опору у відкритому стані. Можуть швидко вмикати та вимикати DC-навантаження. За допомогою широтно-імпульсної модуляції (ШІМ) можна плавно регулювати потужність DC-навантажень (наприклад, яскравість світлодіодів, швидкість DC-двигунів). Низька вартість (окремих компонентів).

Однак, призначені для керування постійним струмом. Відсутність гальванічної ізоляції (якщо не використовувати оптоізоляцію)ю Потрібні додаткові заходи для забезпечення безпеки при керуванні високовольтними DC-навантаженнями. Для керування потужними транзисторами може знадобитися спеціальний драйвер. При значних струмах транзистори можуть сильно нагріватися і потребувати радіаторів.

Ключові критерії вибору засобу керування:

- Тип навантаження (AC/DC).
- Потужність навантаження (струм та напруга).
- Необхідний рівень контролю (ввімкнення/вимкнення, плавне регулювання).
- Частота перемикавання.
- Гальванічна ізоляція.
- Вартість.
- Простота керування з Arduino.

Вибір засобів керування енергоспоживанням для системи на Arduino є важливим етапом, який визначає функціональність, надійність та безпеку

системи. Ретельний аналіз типу та характеристик керованих навантажень, необхідного рівня контролю, вимог до безпеки та бюджету проекту дозволить обрати оптимальні компоненти для ефективного керування енергоспоживанням. Для простих завдань ввімкнення/вимкнення AC та DC навантажень часто використовуються електромеханічні реле та SSR відповідно. Для плавного регулювання AC застосовуються симістори/тріаки або диммери, а для DC - транзистори з ШІМ. Мені необхідно реалізувати просте ввімкнення/вимкнення AC навантажень, тому, з огляду на вартість пристрою, буду використовувати електро-механічне реле.

3 ПРОЕКТУВАННЯ СИСТЕМИ КОНТРОЛЮ ТА КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

3.1 Апаратна реалізація системи

Відповідно до огляду технічних засобів, що буди розглянути у розділу 2 моєї кваліфікаційної роботи система будується з таких основних компонентів, як керуючий пристрій (мікроконтролер), сенсори (датчик струму), актуатори (релейні модулі) та засоби індикації процесів (LCD екран).

В якості датчика струму використовується ACS712[8] , це є популярним рішенням для безконтактного вимірювання постійного та змінного струму. Його вибір для мого проекту обумовлений такими характеристиками:

- ACS712 використовує ефект Холла для вимірювання струму. Це означає, що він вимірює магнітне поле, створюване струмом, що протікає через провідник, без прямого електричного контакту з вимірюваним ланцюгом. Це забезпечує гальванічну розв'язку, що є критично важливим для безпеки, особливо при роботі з високими напругами (наприклад, 220 В змінного струму).

- Завдяки гальванічній розв'язці, низьковольтна частина схеми (Arduino) повністю ізольована від високовольтної частини, що значно знижує ризик ураження електричним струмом для розробника та користувача.

- ACS712 має простий аналоговий вихід, який прямо пропорційний вимірюваному струму. Це дозволяє легко підключити його до аналогового входу мікроконтролеру та зчитувати дані за допомогою стандартних функцій АЦП.

- Датчики ACS712 є дуже доступними та широко поширеними, що сприяє зниженню загальної вартості проекту.

- Датчики ACS712 доступні в різних модифікаціях з різними діапазонами вимірювання струму (наприклад, ± 5 А, ± 20 А, ± 30 А). Це

дозволяє вибрати відповідний датчик залежно від очікуваних значень струму в системі, що контролюється. Для побутових приладів або невеликих навантажень ± 5 А може бути достатньо, тоді як для більш потужних пристроїв знадобляться модифікації на ± 20 А або ± 30 А. (Для макету я вибіру датчик на 5 А, щоб забезпечити максимальну точність).

- ACS712 може вимірювати як постійний, так і змінний струм, що робить його універсальним для більшості систем контролю енергоспоживання, де присутні обидва типи навантажень.

- Для базових завдань моніторингу енергоспоживання точність ACS712 є цілком достатньою. Хоча він може мати деякий шум або похибки, пов'язані з температурною стабільністю, для загального контролю та виявлення аномалій його характеристик вистачає.

В якості мікроконтролера було обрано ATmega328 на базі пристрою Arduino Nano. Це є однією з найпопулярніших мікроконтролерних плат на базі мікроконтролера ATmega328P. Її вибір для мого проекту обґрунтовується наступними перевагами:

- Arduino Nano має дуже невеликі розміри, що робить її ідеальною для вбудованих систем, де простір обмежений. Це дозволяє легко інтегрувати її в корпуси пристроїв або розподільні щитки без значного збільшення габаритів кінцевого рішення.

- Плати Arduino Nano є одними з найдешевших мікроконтролерних рішень на ринку. Це робить проект економічно вигідним, що особливо важливо для освітніх проектів, прототипів або невеликих індивідуальних рішень.

- Хоча ATmega328P є 8-бітним мікроконтролером з відносно невисокою тактовою частотою (16 МГц), його продуктивності цілком достатньо для періодичного зчитування даних з датчика струму, виконання простих обчислень (наприклад, переведення показань струму в потужність при відомій напрузі) та виведення інформації на дисплей. Для локального моніторингу та нескладного керування це оптимальний вибір.

- Датчик струму ACS712 має аналоговий вихід, який легко підключається до аналогових входів Arduino Nano. Вбудований аналого-цифровий перетворювач (АЦП) Arduino забезпечує достатню точність для вимірювання струму в побутових та невеликих промислових системах.

Для керування подачею живлення до споживачів у системі контролю енергоспоживання обираємо електромеханічне реле [10]. Його вибір обґрунтовується наступними факторами:

- Реле забезпечує повну електричну ізоляцію між низьковольтною керуючою частиною (Arduino Nano) та високовольтним силовим ланцюгом. Це є критично важливим для безпеки, оскільки запобігає передачі небезпечної напруги з силового ланцюга на мікроконтролер та інші чутливі компоненти.

- Електромеханічні реле розраховані на комутацію значних струмів (наприклад, 10 А, 16 А, 30 А) та напруг (220 В змінного струму або вище), що робить їх придатними для керування більшістю побутових та офісних електроприладів.

- Керування електромеханічним реле з мікроконтролера є відносно простим. Зазвичай для цього потрібен один цифровий вивід Arduino та транзисторний ключ (або спеціальний модуль реле), який забезпечує достатній струм для спрацьовування котушки реле.

- Електромеханічні реле є економічно вигідним рішенням для комутації навантажень, особливо порівняно з твердотільними реле (SSR) з аналогічними характеристиками струму. - Для завдань увімкнення/вимкнення приладів, які не потребують дуже частої комутації, електромеханічні реле демонструють достатню надійність.

Для відображення інформації я зупинився на LCD 1602 – це дворядковий рідкокристалічний дисплей, здатний відображати 16 символів у кожному рядку. Модифікація з інтерфейсом I2C значно спрощує його підключення. Стандартний LCD 1602 вимагає близько 12-16 проводів для підключення до мікроконтролера, що значно ускладнює монтаж та займає

багато цінних виводів Arduino Nano. Модуль з I2C адаптером вирішує цю проблему, дозволяючи підключити дисплей всього чотирма проводами (VCC, GND, SDA, SCL). Це звільняє більшість виводів Arduino для підключення інших датчиків або виконавчих пристроїв.

LCD 1602 забезпечує чітке текстове відображення інформації, що є достатнім для виведення основних параметрів енергоспоживання (наприклад, поточний струм, потужність, спожита енергія). Символьний дисплей легко читається в більшості умов освітлення. Модулі LCD 1602 з I2C адаптером є дуже поширеними та недорогими, що робить їх привабливим рішенням для бюджетних проектів. Крім того такі модулі мають вбудоване підсвічування, яке можна програмно вмикати або вимикати. Це забезпечує хорошу видимість інформації як у світлих, так і в темних приміщеннях.

Дисплей дозволяє в реальному часі бачити поточні показники енергоспоживання, що є важливим елементом для інформування та стимулювання до енергозбереження.

Комбінація Arduino Nano, ACS712, LCD 1602 I2C та електромеханічного реле створює гармонійну та ефективну платформу для розробки базової системи контролю енергоспоживання з функцією керування. Arduino Nano виступає як центральний процесор, що зчитує аналогові дані струму з ACS712, виконує необхідні обчислення. На основі цих даних або за заданим розкладом, Arduino може активувати електромеханічне реле для увімкнення або вимкнення підключеного споживача. Оброблену інформацію виводиться на LCD 1602 I2C. Усі чотири компоненти є недорогими, що дозволяє створити функціональну систему контролю енергоспоживання з мінімальними фінансовими витратами. Завдяки простоті підключення (особливо I2C дисплея) та широкій підтримці в екосистемі Arduino, розробка та налагодження системи є відносно швидкими та легкими. Гальванічна розв'язка ACS712 та електромеханічного реле забезпечує безпеку роботи з високовольтними ланцюгами, що є критично важливим для будь-якої системи, що взаємодіє з електромережею.

Хоча обрані компоненти є чудовим вибором для базової системи контролю енергоспоживання, важливо розуміти їхні обмеження - ACS712 вимірює лише струм. Для точного вимірювання активної потужності в мережах змінного струму необхідно також вимірювати напругу та фазовий кут між струмом та напругою. Для цього потрібні більш складні датчики (наприклад, PZEM-004T або спеціалізовані мікросхеми енергообліку). Проте, для багатьох побутових навантажень, де коефіцієнт потужності близький до одиниці, або для оцінки споживання, вимірювання лише струму є достатнім. Базова конфігурація не передбачає підключення до Інтернету або локальної мережі, що обмежує можливості віддаленого моніторингу та зберігання даних у хмарі. Для цього потрібно додати модулі Wi-Fi (наприклад, ESP8266) або Ethernet. LCD 1602 є текстовим дисплеєм і не дозволяє відображати графіки або складні візуалізації.

Незважаючи на ці обмеження, обрані компоненти є чудовою відправною точкою для розробки своєї системи контролю енергоспоживання. Вони дозволяють швидко створити функціональний прототип, перевірити основні ідеї та отримати цінний досвід. У майбутньому систему можна розширити, додавши більш досконалі датчики, модулі зв'язку та програмні функції для підвищення її можливостей.

Схема підключення елементів до мікроконтролера:

- Вхід OUT датчика ACS712 – вхід A0 Arduino Nano;
- Вхід S реле – вхід D13 Arduino Nano;
- Вхід SDA LCD – вхід A4 Arduino Nano;
- Вхід SCL LCD – вхід A5 Arduino Nano.

Загальна схема підключення елементів наведено на рис 3.1.

Вибір Arduino Nano, датчика струму ACS712, дисплея LCD 1602 I2C та електромеханічного реле для системи контролю енергоспоживання є обґрунтованим рішенням для проектів, що вимагають простоти, доступності та достатньої функціональності для базового моніторингу, відображення даних та керування. Ця комбінація забезпечує безпеку, легкість інтеграції та

швидку розробку, що робить її ідеальною для освітніх цілей, прототипів або невеликих локальних систем енергомоніторингу та керування. Вона дозволяє ефективно вирішити завдання обліку, візуалізації та базового керування енергоспоживанням, створюючи основу для подальшого вдосконалення та розширення функціоналу СКЕ.

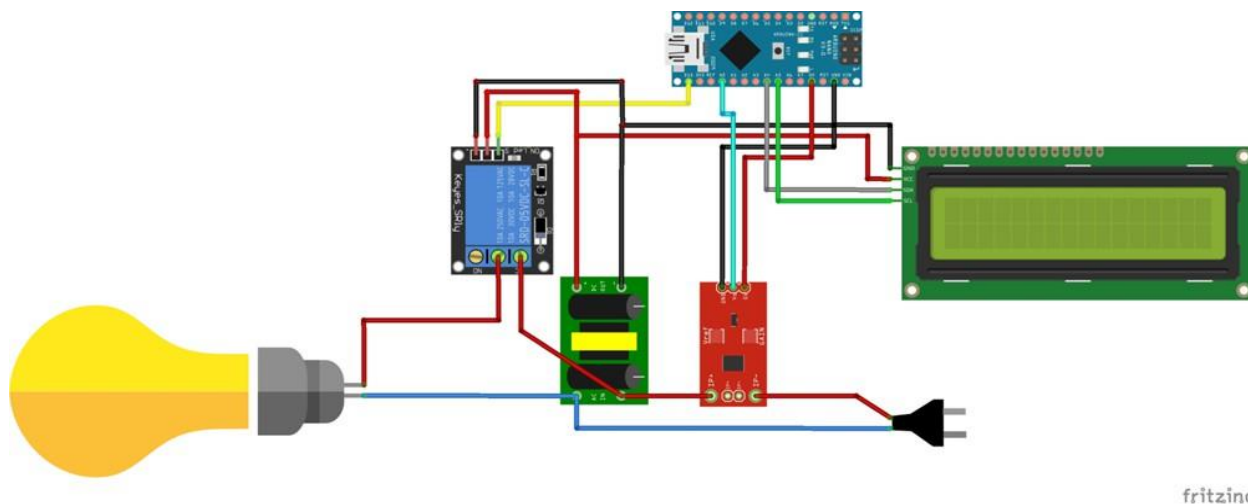


Рисунок 3.1 – Схема системи контролю та керування енергоспоживанням

3.2 Програмна реалізація

Програма для Arduino, як правило, складається з кількох основних частин:

1. Підключення бібліотек (`#include`): На початку скетчу підключаються необхідні бібліотеки, які надають функції для роботи з конкретними апаратними компонентами (наприклад, для LCD дисплея).
2. Глобальні змінні та константи: Оголошуються змінні, доступні з будь-якої частини програми, а також константи, що визначають параметри системи (наприклад, пін для датчика, калібрувальні значення).
3. Функція `setup()`: Ця функція виконується один раз при запуску або перезавантаженні Arduino. Вона використовується для ініціалізації периферійних пристроїв, встановлення початкових значень та налаштування

портів.

4. Функція `loop()`: Ця функція виконується безперервно після завершення `setup()`. Вона містить основну логіку програми: зчитування даних, обчислення, виведення на дисплей та керування.

5. Допоміжні функції (опціонально): Для покращення читабельності та модульності коду, складніші операції можуть бути винесені в окремі функції.

Загальний код програми для мікроконтролера наведено у додатку А (лістінг 3.1). Опис коду

Підключення бібліотек:

- `#include <Wire.h>` Ця бібліотека є стандартною для Arduino і забезпечує функціональність для зв'язку по протоколу I2C (Inter-Integrated Circuit). LCD 1602 з I2C адаптером використовує цей протокол для обміну даними з мікроконтролером. Без неї взаємодія з дисплеєм неможлива.

- `#include <LiquidCrystal_I2C.h> [11]` Це спеціалізована бібліотека, яка спрощує роботу з LCD дисплеями, підключеними через I2C адаптер. Вона надає високорівневі функції, такі як `lcd.init()`, `lcd.print()`, `lcd.setCursor()`, `lcd.clear()` та інші, що дозволяють легко керувати дисплеєм без необхідності працювати безпосередньо з I2C регістрами.

Глобальні змінні та константи:

- `const int ACS_PIN = A0;` Визначає аналоговий пін Arduino, до якого підключений вихід (OUT) датчика ACS712. Вибір A0 є типовим, але можна використовувати будь-який інший аналоговий пін.

- `const int SENSOR_OFFSET = 512;` Це критично важлива константа для калібрування датчика ACS712 [9]. Коли через датчик не протікає струм (0 A), його аналоговий вихід має бути на рівні половини напруги живлення (наприклад, 2.5 В для 5В живлення). При 10-бітному АЦП Arduino (0-1023) це відповідає значенню близько 512. Точне значення може трохи відрізнятись для кожного екземпляра датчика та залежить від стабільності живлення. Для підвищення точності можливо провести калібрування - підключити датчик до

Arduino, не підключаючи до нього навантаження, і зчитати середнє значення з ACS_PIN. Отримане значення слід використовувати як SENSOR_OFFSET.

- const float SENSITIVITY = 185.0; Визначає чутливість датчика ACS712- 05A у мілівольтах на Ампер (mV/A). Це значення є специфічним для моделі датчика

- oACS712-05A: 185 mV/A

- oACS712-20A: 100 mV/A

- oACS712-30A: 66 mV/A.

- const float NOMINAL_VOLTAGE = 220.0; Номінальна напруга електричної мережі, для якої розраховується потужність. Для України це 220 В. Ця константа використовується для простого розрахунку потужності за формулою $P=U \times I$.

- const int LCD_ADDR = 0x27; I2C адреса адаптера LCD дисплея. (Якщо дисплей не працює, можливо, потрібно спробувати 0x3F або 0x30)..

- const int LCD_COLS = 16; та const int LCD_ROWS = 2; Визначають розмір дисплея (16 символів у 2 рядках).

- LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS); Створює об'єкт lcd класу LiquidCrystal_I2C, який буде використовуватися для взаємодії з дисплеєм.

- float instantaneousCurrent = 0.0;, float rmsCurrent = 0.0;, float instantaneousPower = 0.0;, float averagePower = 0.0; Змінні для зберігання вимірянних та розрахованих значень струму та потужності. Використання float дозволяє працювати з дробовими значеннями.

- const int NUM_READINGS = 50, readings[NUM_READINGS], int readIndex = 0, int averageRaw = 0; long total = 0; Ці змінні використовуються для реалізації простого алгоритму згладжування (усереднення) показань АЦП. Це допомагає зменшити шум та стабілізувати відображення на дисплеї.

- const int SAMPLES = 1000;, long sumOfSquares = 0; Ці змінні використовуються для розрахунку середньоквадратичного (RMS) значення струму, що є більш репрезентативним для змінного струму, ніж миттєве

значення.

Функція `setup()`

Функція проводить початкову ініціалізацію елементів системи.

- `Serial.begin(9600);` Ініціалізує послідовний порт для зв'язку з комп'ютером.

- `lcd.init();` Ініціалізує I2C адаптер дисплея.

- `lcd.backlight();` Вмикає підсвічування дисплея.

- `if (lcd.begin(LCD_COLS, LCD_ROWS)) { ... } else { ... }` – Перевіряє, чи успішно ініціалізовано дисплей. Якщо так, виводить привітання на дисплей та монітор порту. Якщо ні, виводить повідомлення про помилку на монітор порту.

- Цикл `for` – Заповнює масив `readings` нулями, щоб уникнути "сміття" при перших обчисленнях середнього значення.

Функція `loop()`

Ця функція є постійно виконуємою частиною програми, де відбувається безперервний цикл вимірювань, обчислень та виведення даних.

Зчитування та згладжування даних

`int rawValue = analogRead(ACS_PIN);` Зчитує поточне аналогове значення з датчика ACS712.

Блок коду з `total`, `readings`, `readIndex`, `averageRaw` – Реалізує алгоритм "ковзного середнього". Це означає, що програма зберігає останні `NUM_READINGS` вимірювань і постійно обчислює їхню суму та середнє. Це допомагає згладити короткочасні коливання та шум, роблячи показання на дисплеї більш стабільними.

Розрахунок миттєвого струму

`instantaneousCurrent = ((float)rawValue - SENSOR_OFFSET) * (5000.0 / 1024.0) / SENSITIVITY;`

`(float)rawValue - SENSOR_OFFSET` – Віднімаємо зміщення, щоб отримати "чисте" відхилення сигналу від нульового струму.

`* (5000.0 / 1024.0)` – Перетворює сире 10-бітне значення АЦП (0-1023)

у мілівольти. 5000.0 - це 5В у мілівольтах, 1024.0 - максимальне значення АЦП.

/ SENSITIVITY – Ділить отриману напругу в мВ на чутливість датчика (мВ/А), щоб отримати струм в Амперах.

Розрахунок RMS струму (середньоквадратичного значення)

Для змінного струму миттєве значення постійно змінюється. RMS значення є більш показовим для визначення ефективності струму.

Цикл `for (int i = 0; i < SAMPLES; i++) { ... }` – Програма швидко збирає SAMPLES (1000) вимірювань з датчика.

`int centeredSample = sample - SENSOR_OFFSET;` Кожне значення центруються відносно нульового зміщення.

`sumOfSquares += (long)centeredSample * centeredSample;` Обчислюється сума квадратів цих центрованих семплів. Використання `long` для `sumOfSquares` запобігає переповненню, оскільки квадрати можуть бути великими.

`delayMicroseconds(100);` Невелика затримка між зчитуваннями, щоб дати АЦП час на конвертацію.

`float meanSquare = (float)sumOfSquares / SAMPLES;` Обчислюється середнє квадратичне значення.

`float rmsVoltageSensor = sqrt(meanSquare);` Береться квадратний корінь, щоб отримати RMS значення напруги, яку видає датчик.

`rmsCurrent = rmsVoltageSensor * (5000.0 / 1024.0) / SENSITIVITY;` Перетворює RMS напругу датчика в RMS струм, використовуючи ті ж коефіцієнти, що й для миттєвого струму.

Розрахунок потужності

`instantaneousPower = abs(instantaneousCurrent) * NOMINAL_VOLTAGE;`

Розраховує миттєву потужність як добуток абсолютного значення миттєвого струму на номінальну напругу. Використання `abs()` (абсолютного значення) гарантує, що потужність завжди буде позитивною.

`averagePower = rmsCurrent * NOMINAL_VOLTAGE;` Розраховує

середню потужність як добуток RMS струму на номінальну напругу. Для побутових приладів це часто є більш репрезентативним показником, ніж миттєва потужність.

Виведення даних на послідовний порт

Блок `Serial.print()` та `Serial.println()`: Виводить усі розраховані значення на монітор послідовного порту. Це незамінний інструмент для налагодження, оскільки дозволяє бачити, які значення отримує та обробляє мікроконтролер. `Serial.print(value, N)` виводить значення з N знаками після коми.

Виведення даних на LCD дисплей

`lcd.clear()`; Очищає весь дисплей. Це важливо, щоб уникнути "накладання" символів, якщо нові значення коротші за попередні.

`lcd.setCursor(0, 0)`; Встановлює курсор у першу позицію першого рядка (стовпець 0, рядок 0).

`lcd.print("I: "); lcd.print(rmsCurrent, 2); lcd.print(" A");` Виводить текст "I: " (струм), потім значення `rmsCurrent` з двома знаками після коми, і потім " A" (Ампери).

`lcd.setCursor(0, 1)`; Встановлює курсор у першу позицію другого рядка (стовпець 0, рядок 1).

`lcd.print("P: "); lcd.print(averagePower, 0); lcd.print(" W");` Виводить текст "P: " (потужність), потім значення `averagePower` без знаків після коми, і потім " W" (Ватти).

`delay(500)`; Затримка на 500 мілісекунд (0.5 секунди). Ця затримка контролює частоту оновлення дисплея. Занадто швидке оновлення може зробити показання нечитабельними, а занадто повільне – неактуальними.

В цілому розроблена програма для Arduino Nano, що використовує датчик ACS712 та дисплей LCD 1602 I2C, є надійною основою для створення базової системи контролю енергоспоживання. Вона демонструє основні принципи збору, обробки та відображення даних про струм та потужність. Завдяки простоті та доступності обраних компонентів, цей проект може виступати основою домашньої автоматизації. При подальшим розширенням

функціоналу, ця система може бути адаптована для вирішення більш складних завдань енергоефективності.

Ця базова програма може бути значно розширена для створення більш функціональної системи контролю. Можна додати модуль вимірювання напруги (наприклад, резистивний дільник або спеціалізований датчик), що дозволить вимірювати реальну напругу в мережі, а не використовувати номінальне значення.

Здійснити розрахунок активної потужності та коефіцієнта потужності. Для цього потрібно одночасно вимірювати миттєві значення струму та напруги, а потім виконувати складніші обчислення (наприклад, інтегрування добутку струму та напруги за період). Це вимагає більш потужного мікроконтролера або спеціалізованої мікросхеми енергообліку (наприклад, ADE7753, HLW8012).

Додавання модуля SD-карти дозволить зберігати виміряні дані для подальшого аналізу та побудови графіків. Інтеграція модуля Wi-Fi (наприклад, ESP8266 або ESP32) дозволить відправляти дані на хмарні платформи (наприклад, ThingSpeak, Blynk, Firebase) для віддаленого моніторингу, зберігання та візуалізації. Додавання кнопок або енкодера дозволить користувачеві взаємодіяти з системою, перемикати режими відображення, встановлювати пороги тощо.

При використанні Wi-Fi модуля можна створити просту веб-сторінку, доступну з будь-якого пристрою в мережі, для відображення даних та керування. OLED дисплеї (наприклад, 128x64) можуть відображати графіки та більш складну інформацію, ніж LCD 160.

ВИСНОВКИ

Створення системи контролю та управління енергоспоживанням на базі мікроконтролерної плати Arduino Nano, датчика струму ACS712, рідкокристалічного дисплея LCD 1602 I2C та електромеханічного реле є важливим кроком у розумінні та практичному застосуванні принципів енергоефективності. Цей проект демонструє, як за допомогою доступних та відносно недорогих компонентів можна реалізувати функціональне рішення для моніторингу та базового управління електричними навантаженнями. Протягом реалізації проекту були отримані цінні знання та досвід у галузі апаратного та програмного забезпечення, що дозволяє зробити наступні висновки.

Вибір Arduino Nano, ACS712, LCD 1602 I2C та стандартного реле дозволив створити функціональний прототип з мінімальними фінансовими витратами. Це робить проект ідеальним для освітніх цілей, хобі-проектів або як початкова точка для стартапів з обмеженим бюджетом.

Система забезпечує безперервне зчитування даних про струм за допомогою ACS712 та розрахунок потужності. Відображення цих даних на LCD дисплеї в реальному часі надає користувачеві негайний зворотний зв'язок про поточне енергоспоживання пристрою, що контролюється.

Інтеграція електромеханічного реле дозволяє системі не тільки моніторити, але й активно впливати на енергоспоживання, вмикаючи або вимикаючи підключене навантаження. Це відкриває можливості для реалізації простих сценаріїв автоматизації, наприклад, вимкнення пристрою при перевищенні певного порогу потужності або за розкладом.

Використання датчика ACS712 та електромеханічного реле забезпечує гальванічну розв'язку між високовольтним силовим ланцюгом та низьковольтною керуючою електронікою Arduino. Це є критично важливим аспектом безпеки, що захищає мікроконтролер та користувача від небезпечних напруг.

Проект є чудовим практичним прикладом для вивчення основ електроніки, мікроконтролерів, аналогово-цифрового перетворення, роботи з датчиками та дисплеями, а також базових принципів енергомоніторингу та автоматизації.

Під час розробки та тестування системи були виявлені певні виклики, які дозволили отримати важливі уроки:

Датчик ACS712 вимірює лише струм. Для точного розрахунку активної потужності в мережах змінного струму необхідно також вимірювати напругу та враховувати коефіцієнт потужності (фазовий зсув між струмом та напругою). В даному проекті потужність розраховується як добуток струму на номінальну напругу, що дає лише видиму потужність. Це обмеження важливо враховувати при інтерпретації даних.

Точність показань сильно залежить від коректного калібрування датчика ACS712, зокрема визначення точного `SENSOR_OFFSET` (зміщення при нульовому струмі) та `SENSITIVITY` (чутливості). Неправильне калібрування призводить до значних похибок.

Аналогові вимірювання схильні до електричного шуму. Застосування програмних методів згладжування (наприклад, "ковзне середнє" або усереднення RMS) є необхідним для стабілізації показань на дисплеї.

Реле є надійним для нечастої комутації, але має обмежений ресурс за кількістю спрацьовувань, видає клацаючий звук і не підходить для швидких перемикачів або димування. Для таких завдань краще підходять твердотільні реле (SSR).

Розроблена система може знайти застосування в таких сферах, як домашній моніторинг енергоспоживання, моніторинг у малих офісах/майстернях, Освітні та демонстраційні цілі, Основа для прототипування

В цілому розроблений мною проект системи контролю та управління енергоспоживанням на Arduino Nano з датчиком ACS712, дисплеєм LCD 1602 I2C та реле є успішним підтвердженням концепції. Він демонструє

можливість створення ефективних, доступних та безпечних рішень для моніторингу та базового управління енергетичними навантаженнями. Незважаючи на певні обмеження, властиві базовим компонентам, система має значний потенціал для подальшого розвитку та масштабування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасні автоматизовані системи контролю та обліку енергоресурсів. [Електроний ресурс]. – Режим доступу: <http://energetika.in.ua/ua/books/book-5/part-4/section-6>
2. Енергоефективні заходи в громадах України: кроки до сталого розвитку та енергетичної незалежності у 2025 році. [Електроний ресурс]. – Режим доступу: <https://onova.org.ua/news/enerhoefektyvni-zakhody-v-hromadakh-ukrainy-kroky-do-staloho-rozvytku-ta-enerhetychnoi-nezalezhnosti-u-2025-rotsi>
3. Energy Management Systems – A Complete Guide (Mavarick AI). [Електроний ресурс]. – Режим доступу: <https://mavarick.ai/blogs/energy-management-systems-complete-guide/>
4. What is EMS? Meaning, Benefits & Features Explained! (Electro Industries/GaugeTech) Електроний ресурс]. – Режим доступу:– <https://www.electroind.com/a-comprehensive-guide-to-energy-management-control-system/>
5. Home Energy Monitoring with Arduino (All About Circuits) [Електроний ресурс]. – Режим доступу: <https://www.instructables.com/Simple-Arduino-Home-Energy-Meter/>
6. Arduino Forum. [Електроний ресурс]. – Режим доступу: – <https://forum.arduino.cc/>
7. Arduino Nano Pinout, Specs, and How to Use It. [Електроний ресурс]. – Режим доступу: <https://docs.arduino.cc/hardware/nano/>
8. ACS712 Datasheet (Allegro MicroSystems). [Електроний ресурс]. – Режим доступу: <https://www.alldatasheet.com/datasheet-pdf/pdf/168326/ALLEGRO/ACS712.html>
9. ACS712 Current Sensor Calibration. Електроний ресурс]. – Режим доступу: <https://forum.arduino.cc/t/how-to-calibrate-ac712-properly/540323>

10. Arduino Relay Tutorial (Random Nerd Tutorials)/ [Электроний ресурс]. – Режим доступа: <https://arduinogetstarted.com/tutorials/arduino-relay>
11. LiquidCrystal_I2C Library GitHub. [Электроний ресурс]. – Режим доступа: https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library/blob/master/LiquidCrystal_I2C.h

ДОДАТОК А

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int ACS_PIN = A0;
const int SENSOR_OFFSET = 512; const
float SENSITIVITY = 185.0;
const float NOMINAL_VOLTAGE = 220.0; const
int LCD_ADDR = 0x27;
const int LCD_COLS = 16; const
int LCD_ROWS = 2;

LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

float instantaneousCurrent = 0.0;
float rmsCurrent = 0.0;
float instantaneousPower = 0.0;
const int NUM_READINGS = 50; int
readings[NUM_READINGS];
int readIndex = 0; long
total = 0;
int averageRaw = 0;
const int SAMPLES = 1000; long
sumOfSquares = 0;

void setup() {
  Serial.begin(9600);
  Serial.println("Система контролю енергоспоживання
запускається...");
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Start SEC");
  lcd.setCursor(0, 1);
  lcd.print("please wait");

  for (int i = 0; i < NUM_READINGS; i++) { readings[i] = 0;
}
}

void loop() {

```

Лістинг 3.1 - Код програми для мікроконтролера

```

int rawValue = analogRead(ACS_PIN);
total = total - readings[readIndex];
readings[readIndex] = rawValue; total = total +
readings[readIndex];
readIndex = readIndex + 1;
if (readIndex >= NUM_READINGS) { readIndex = 0;
}
averageRaw = total / NUM_READINGS;
instantaneousCurrent = ((float)rawValue - SENSOR_OFFSET) *
(5000.0 / 1024.0) / SENSITIVITY;
sumOfSquares = 0;
for (int i = 0; i < SAMPLES; i++) { int
sample = analogRead(ACS_PIN);
int centeredSample = sample - SENSOR_OFFSET ;
sumOfSquares += (long)centeredSample * centeredSample;
delayMicroseconds(100); }
float meanSquare = (float)sumOfSquares / SAMPLES; float
rmsVoltageSensor = sqrt(meanSquare);
rmsCurrent = rmsVoltageSensor * (5000.0 / 1024.0) /
SENSITIVITY;
instantaneousPower = abs(instantaneousCurrent) * NOMINAL_VOLTAGE;
averagePower = rmsCurrent * NOMINAL_VOLTAGE;
Serial.print("Сире значення АЦП: ");
Serial.print(rawValue); Serial.print(" | Згладжене сире: ");
Serial.print(averageRaw); Serial.print(" | Миттєвий струм: ");
Serial.print(instantaneousCurrent, 3); // 3 знаки після коми
Serial.print(" A | RMS струм: ");
Serial.print(rmsCurrent, 3);
Serial.print(" A | Миттєва потужність: ");
Serial.print(instantaneousPower, 2); // 2 знаки після коми
Serial.print(" Вт | Середня потужність: ");
Serial.print(averagePower, 2);
Serial.println(" Вт");
lcd.clear(); lcd.setCursor(0, 0); lcd.print("I: ");
lcd.print(rmsCurrent, 2); lcd.print(" A"); lcd.setCursor(0, 1);
lcd.print("P: ");
lcd.print(averagePower, 0); lcd.print(" W"); delay(500);
}

```

Лістинг 3.1, аркуш 2