

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Модель імплементації технології "Energy harvesting"
в якості альтернативного джерела енергії у вбудованих
системах
(тема)

Виконав: студент 2 курсу, групи СКСм-21-1

Гелетто В.М.
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія та управління
(код і повна назва спеціальності)


Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник Філіппенко І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри


(підпис)

Чумаченко С.В.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Автоматизації проектування обчислювальної техніки _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 Комп'ютерна інженерія _____
(шифр і назва)

Тип програми _____ Освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Спеціалізовані комп'ютерні системи _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Гелетто Віолетті Максимівні _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Модель імплементації технології "Energy harvesting" в якості
альтернативного джерела енергії у вбудованих системах _____

затверджена наказом університету від _____ 14 _____ листопада _____ 2022 р. № _____ 1478Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 2022 р.

3. Вихідні дані до роботи _____
технологія "Energy harvesting"
вбудовані системи _____

4. Перелік питань, що потрібно опрацювати в роботі _____
Обґрунтування актуальності роботи _____

Аналіз предметної області та постановка задачі _____
Розробка системи спостереження за станом здоров'я людини з використанням технології
energy harvesting _____

Програмна та апаратна реалізація _____
Аналіз отриманих результатів _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____
24 слайди _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	04.11.2022 - 05.11.2022	
2	Аналіз літератури	12.11.2022 - 16.11.2022	
3	Розробка моделі	17.11.2022 - 26.11.2022	
4	Реалізація алгоритму оптимізації	29.11.2022 - 05.12.2022	
5	Тестування отриманих даних	06.12.2022 - 10.12.2022	
6	Оформлення пояснювальної записки	13.12.2022 - 17.12.2022	

Студент _____
(підпис)

Керівник роботи _____ доц. Філіппенко І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить 59 с., 12 рис., 6 табл., 3 дод., 29 джерел.

EMBEDDED SYSTEMS, ENERGY HARVESTING, АЛЬТЕРНАТИВНІ ДЖЕРЕЛА ЕНЕРГІЇ, МІКРОКОНТРОЛЕР, АРХІТЕКТУРА

Метою кваліфікаційної роботи є розробка системи спостереження за станом здоров'я людини на основі вбудованих систем з використанням, в якості альтернативного джерела енергії, технології «energy harvesting».

Для цього в роботі було досліджено потреби системи в живленні та проведено порівняння кількості необхідної та кількості енергії, що може дати система «energy harvesting». Для цього було проведено дослідження кожного з компонентів системи. А після цього проведено аналіз отриманих даних та застосування інформації відносно розробки реальної моделі.

Прототип моделі був реалізований на основі плати Seeeduno Mega, що базується на ATmega 2560. Було реалізоване програмне забезпечення для системи відстеження стану пацієнта з використанням методу оптимізації енергоспоживання.

Результатом кваліфікаційної роботи є модель спостереження за станом здоров'я людини, що може в якості джерела енергії використовувати технології «energy harvesting». Модель базується на принципах побудови вбудованих систем, та застосовує концепцію інтернету речей для передачі даних від пристрою по мережі.

ABSTRACT

The explanatory note of the qualification work: 59 pages, 12 figures, 6 tables, 3 appendices, 29 sources.

EMBEDDED SYSTEMS, ENERGY HARVESTING, ALTERNATIVE ENERGY SOURCES, MICROCONTROLLER, ARCHITECTURE

The goal of the qualification work is to develop a system for monitoring human health based on embedded systems using «energy harvesting» technology as an alternative source of energy.

For this purpose, the system's power needs were analyzed and a comparing was made between the amount of energy needed and the amount of energy that the «energy harvesting» system can provide. For this, a study of each of the system components was conducted. And after that, the analysis of the obtained data and the application of information regarding the development of a real model was carried out.

The prototype model was implemented on the basis of the Seeeduino Mega microcontroller based on ATmega 2560. The software for the patient monitoring system using the energy consumption optimization method was implemented.

The result of the qualification work is a model for monitoring the state of human health, which can use «energy harvesting» technologies as an energy source. The model is based on the principles of building embedded systems, and applies the concept of the Internet of Things to transfer data from the device over the network.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1 Обґрунтування актуальності роботи	13
1.2 Аналіз існуючих рішень «energy harvesting»	17
1.3 Методи динамічного керування живленням	19
1.4 Постановка задачі	21
2 РОЗРОБКА СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА СТАНОМ ЗДОРОВ'Я ЛЮДИНИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ENERGY HARVESTING	23
2.1 Вибір мікроконтролеру	23
2.2 Вибір датчиків, сенсорів та модулів	26
2.3 Вибір джерела живлення системи	31
2.4 Реалізація моделі	34
2.5 Енергоспоживання приладу	35
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА СТАНОМ ЗДОРОВ'Я	37
3.1 Компоненти реалізованої системи.....	37
3.2 Методи оптимізації енергоефективності системи	37
3.3 Огляд мови програмування розробки.....	38
3.4 Огляд середовища розробки - Arduino IDE.....	38
3.5 Структура програмної частини.....	40
3.6 Використані бібліотеки	43
3.7 Реалізація системи спостереження за станом здоров'я та алгоритму зниження енергоспоживання	45
4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ.....	53
4.1 Аналіз розробленого програмного забезпечення	53
4.2 Результати тестування.....	53

ВИСНОВКИ	56
Перелік джерел посилання.....	57
ДОДАТОК А.....	Ошибка! Закладка не определена.
ДОДАТОК Б	Ошибка! Закладка не определена.
ДОДАТОК В.....	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GUI – grafical user snterface

BC – вбудована система

DSP – digital signal processor

BLE – bluetooth low energy

SoC – system on crystal

PROM – programmable read-only memory

EEPROM – electrically erasable programmable read-only Memory

ПЗ – програмне забезпечення

PZT-кераміка – п'єзокераміка

ККД – коефіцієнт корисної дії

WSN – бездротова сенсорна мережа

IoT – internet of things

MEMS – мікроелектромеханічні системи

EOM – електронно обчислювальна машина

ICSP – внутрішньосхемне програмування

UART – універсальний асинхронний приймач-передавач

ШИМ – широтно-імпульсна модуляція

RISC – reduced instruction set computer

DMA – прямий доступ до пам'яті

ЕКГ – електрокардіографія

АЦП – аналогово-цифровий перетворювач

UVLO – режим блокування низької напруги наднизького струму спокою

IDE – integrated development environment

ВСТУП

Сучасна енергетична та економічна ситуація у світі змушує задуматися над застосуванням альтернативних джерел енергії в більшій кількості. На даний момент, найпоширенішим типом є сонячна енергія та енергія вітру. Дані способи є більш пристосованими до масштабного видобутку енергії. Але якщо подивитися на останні тенденції, можна помітити стрімкий розвиток такого виду альтернативних джерел енергії, як «energy harvesting».

Energy harvesting – це вид отримання енергії від навколишнього середовища. У даному варіанті джерелом живлення може бути безліч явищ природи, що оточують нас. Серед доступних джерел енергії найчастіше використовують світло, різницю температур, електромагнітну або індукційну енергію, кінетичну та радіочастотну енергію.

Використання даного типу альтернативного джерела енергії може поліпшити використання вбудованих систем, а саме відсутність необхідності у дротовому зв'язку та постійної зміни батарей, або заряджання акумуляторів може призвести до кардинального розширення сфери застосування «energy harvesting» у вбудованих системах. Нижче приведені деякі з можливих варіантів застосування «energy harvesting» у вбудованих системах:

- сфера здоров'я. Тут може бути безліч прикладів, один з основних це система відстеження здоров'я, що працює в реальному часі. Вона може застосовуватися для людей, що повинні бути під наглядом постійно (в закладах здоров'я, або в себе дома), також використання таких систем може бути доречно для військових, де потрібно відстежувати стан людини, та помічати, якщо необхідна екстрена медична допомога;

- сфера безпеки життєдіяльності людини. Будь-які датчики вогню, що викликають пожежників, системи виклику поліції, при натисканні кнопки. Використання в системах сигналізацій в машинах, будинках;

- сфера погоди та клімату. Використання можливе в датчиках

термометрів, барометрів, що є частинами метеологічних станцій. Кількість енергії навколо дасть змогу жити не лише пристрій, а й збирати додаткову енергію;

- smart технології. У даній сфері використання мікроконтролерів зростає постійно. При використанні додатково ще «energy harvesting» можливо досягти розповсюдження таких систем;

- аеротехніка. Використання технології у літаках, що оточені електромагнітними хвилями зі всіх сторін, під час польоту надає можливість жити вбудовані в систему літака пристрої.

Основними вимогами до вбудованих систем з використанням «energy harvesting» є:

- малопотужність. Доволі часто, мікроконтролери та датчики, які вони використовують, не потребують великої кількості енергії, але на це, все одно, потрібно звертати увагу. Перед розвитком «energy harvesting» не стоїть завдання забезпечувати енергією великих споживачів. Величина одержуваної енергії варіюється в межах декількох мікрвольт і дуже залежить від методу та розмірів генерованого пристрою;

- надійність. Для збільшення терміну придатності системи, необхідно, щоб усі компоненти мали довгий час служби;

- енергоефективне ПЗ. Саме від нього залежить дуже багато. Можливо вибрати малопотужні компоненти, але якщо не регулювати живлення за допомогою спеціальних алгоритмів, може не вистачити для роботи пристрою отриманої енергії;

На жаль, за допомогою «energy harvesting» не можливо отримати велику кількість енергії, але даний спосіб дуже добре підтримує живлення малопотужних систем, тому їх можна використовувати у вбудованих системах.

Перевагами розробки пристроїв на основі вбудованих систем з використанням «energy harvesting» є:

- можливість економити на електроенергії. При тому, що розповсюдження вбудованих систем наразі набирає обертів, самозабезпечення енергією приладу дає можливість вбудованим системам застосовуватися у

величезній кількості проєктів;

– відсутність дротів, що потрібно проводити та батарей, що потрібно замінювати дає можливість розташовувати пристрій у найвіддаленішому місці, або місцях, де людина не може існувати (дослідження океану, космосу). Також це дозволяє розташовувати системи датчиків у важкодоступних місцях, налаштувавши лише при монтажі, та потім не витратити час на можливі заміни батарей, усунення розривів живильних дротів, тим самим підвищивши надійність пристроїв;

– можливість вбудованої системи повноцінно працювати в режимі реального часу. Підживлення системи відбувається постійно, що надає можливість бути на зв'язку майже 24/7.

Пристрої збору енергії використовують вібрацію оточуючих електромагнітних хвиль, різницю температур і перетворюють її на корисну енергію. Тим самим дають можливість жити малопотужні пристрої майже з нічого. Звичайно, такий метод має також і недоліки. Один з очевидних – це дороговизна матеріалів для виготовлення даних пристроїв отримання енергії.

Мета роботи – дослідження використання технології «energy harvesting» у вбудованих системах, а саме системи спостереження за здоров'ям людини. У якості об'єкту дослідження обрано вбудовані системи. Предмет дослідження – модель імплементації технології «Energy harvesting» в якості альтернативного джерела енергії у вбудованих системах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Обґрунтування актуальності роботи

Швидкий розвиток бездротових технологій та електроніки з низьким енергоспоживанням дозволив ширше використовувати автономні системи реального часу та моніторингу. Нещодавні дослідження були зосереджені на розробці систем, які можуть жити за рахунок отримання енергії від навколишнього середовища, такі як механічні вібрації, сонячне світло і вітер, або отримання електромагнітних хвиль безпосередньо з навколишнього середовища.

Energy harvesting або технологія збору енергії – це процес перетворення енергії навколишнього середовища в електричну енергію для використання в живленні автономних електронних пристроїв. Енергія навколишнього середовища, наприклад сонячна, теплова, п'єзоелектрична, радіочастотна та кінетична, отримується та зберігається для заряджання малопотужних пристроїв, які використовуються в електроніці, яка потребує постійних пересувань, в бездротових сенсорних мережах, у вбудованих системах. Описаний принцип полягає у збиранні та акумулюванні енергії навколишнього середовища та перетворення її в електричну [1].

Провідні компанії використовують системи збору енергії в різних галузях промисловості, таких як споживча електроніка, військова та авіакосмічна промисловість, автомобільна промисловість і охорона здоров'я. Крім того, розробка передових систем збору енергії сприяє індустріалізації технології збору енергії.

У теперішній час, проблеми енергозбереження та енергоефективності стають все більш актуальними. Застосування відновлюваних джерел енергії, зменшення забруднень, пошук більш енергоємних методів і технологій – все це

активно вивчається та розвивається. «Energy harvesting» є одним з перспективних видів альтернативних джерел енергії для малопотужних вбудованих систем та систем реального часу.

Використання «energy harvesting» у вбудованих системах має свої недоліки та переваги. Недоліки полягають в тому, що такий спосіб енергії можна використовувати лише в малопотужних системах, крім того навіть в них краще використовувати ще додатково алгоритми для відстеження кількості енергії та економії споживання в разі її недостатності. До позитивних сторін використання можна віднести можливість встановлювати такі вбудовані системи у важкодоступних місцях, автономність, можливість використовувати в системах реального часу, де важливою частиною є постійна працездатність системи.

Питання енергозбереження у сенсорних системах, здебільшого, представляється у вигляді скорочення споживання енергії за рахунок використання нових методів та алгоритмів. Створення досконаліших пристроїв, здатних економити енергію та споживати меншу її кількість є пріоритетним завданням для вчених та інженерів.

Можна сказати, що використання таких методів разом з використанням технології «energy harvesting» може привести до більш переконливих змін у зручності використання систем реального часу, вбудованих систем, та так званих, носимих систем, що повинні знаходитися на тілі людини майже постійно.

Технологія збору енергії базується на ідеї, що пристрої можуть збирати енергію, присутню в навколишньому середовищі, у режимі реального часу та використовувати її негайно, тому енергію потрібно зберігати лише тимчасово. Це може дозволити вбудованим системам досягти теоретично нескінченного терміну служби, обмеженого лише терміном служби їхніх компонентів.

Вбудовані системи, включаючи бездротові датчики та переносне електронне обладнання, є найпоширенішими сферами застосування методу

збору енергії. Очікується, що до 2024 року світовий ринок пристроїв, що живляться від навколишньої енергії, становитиме 2,6 мільярди одиниць. Однак збір енергії створює низку нових проблем, більшість з яких можна віднести до неконтрольованого та непередбачуваного характеру більшості джерел навколишньої енергії. Тому є необхідним використовувати програму для відстеження та зменшення використання енергії у разі потреби. Тоді системи, що базуються на мікроконтролерах зможуть мати перевагу перед іншими методами живлення.

Очевидними мінусами такого підходу насамперед є складність виготовлення таких пристроїв, а разом з цим і їхня вартість. В даний час матеріали, здатні до генерації електричної енергії в тому вигляді і величині, яка потрібна, доволі дорогі і складні у своєму синтезі. Складність пристроїв «energy harvesting» доволі висока – матеріали, як правило, багат шарової структури, для котрих потрібне складне та докладне моделювання.

Приклад можливого застосування пристрою з елементом, здатним генерувати електричну енергію, є система спостереження за станом здоров'я пацієнта. Багат шарові структури кристалів PZT-кераміки, що можуть знаходитися в підшві взуття, при ходьбі можуть використовувати механічну деформацію, в якості живлення носимої системи [6]. Така система підійде для пацієнтів, яким потрібен контроль за самопочуттям протягом тривалого часу - наприклад, люди з обмеженими можливостями, самотні старі люди.

Програмне забезпечення такої системи повинно володіти можливістю централізованої відправки даних до будь-яких контролюючих центрів. Для того, щоб можна було запобігати небезпечним ситуаціям для таких людей, направляти до них бригади швидкої допомоги або пропонувати профілактичне відвідування фахівців охорони здоров'я. Такі системи вже багаторазово реалізовані, однак, продовжують бути залежними від елементів живлення, які потрібно вчасно заряджати або змінювати. Таким чином, вирішення проблеми забезпечення електроживленням може бути у застосуванні технології «energy

harvesting». Звісно, не можна говорити про повну автономність, притому, як уже було сказано вище, вартість таких пристроїв буде доволі висока для масового застосування, але у перспективі можливість використання такого підходу виглядає переконливо.

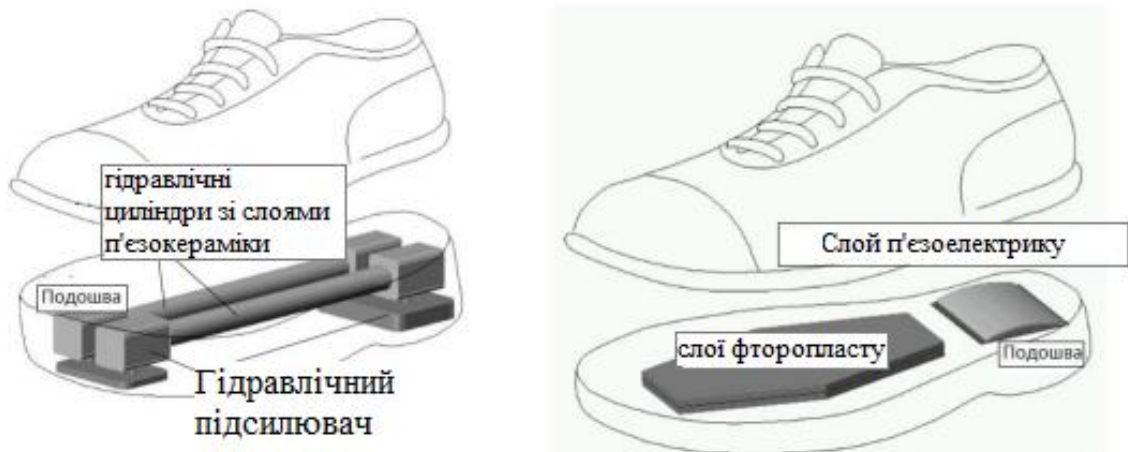


Рисунок 1.1 – Варіант реалізації п'єзоелектричних генераторів, які розташовані у підшві взуття [6].

Використання матеріалів з властивостями отримання енергії з оточуючого середовища можна застосувати майже в кожній сфері людської діяльності. Деякі виробники електроніки вже зараз пропонують свої типові рішення для реалізації технології самозабезпечення енергією. Готові мікросхеми, сенсори, здатні до генерації електричної енергії – це все можливо придбати і використовувати при вирішенні абсолютно різних завдань.

Таким чином, можна зробити загальний висновок про перспективність застосування технології «energy harvesting» у вбудованих системах:

– значення електричної енергії, що генерується, невеликі, проте цілком достатні для живлення малопотужних пристроїв, що базується на мікроконтролерах [3];

– разом з цим з'являється проблема, при якій чим енерговитратніший пристрій, тим більша кількість матеріалу потрібна для його живлення. Звідси виникає необхідність у створенні більш ефективних в енергетичному плані алгоритмів роботи вбудованих систем;

– автономність пристроїв все ж таки обмежена, проте в деяких пристроях цього достатньо, щоб вся система працювала ефективно і надійно;

– вартість матеріалів залишається доволі високою, проте сучасні технології моделювання та синтезу прагнуть зменшити вартість виробництва та застосування;

– енергія, що перетворюється на електричну (вібрації, зміни температури і т.д.), легкодоступна та розповсюджена.

Реалізація вбудованих систем з використанням технології «energy harvesting» все ще залишається доволі вимогливою зі сторони енерговитрат. Це є підставою для реалізації алгоритму, що може зменшити енерговитрати вбудованої системи та максимально оптимізувати енергоефективність моделі відстеження стану здоров'я пацієнта на основі мікроконтролера.

1.2 Аналіз існуючих рішень «energy harvesting»

Впродовж усієї історії, людство намагалося знайти «вічний двигун», або найменш затратні джерела енергії з максимальним ККД. Тому не дивно, що так багато ідей для створення таких механізмів. На даний момент, існує дуже багато різновидів методу «energy harvesting». Усі вони не можуть забезпечити енергією у промислових масштабах, але доволі ефективні для використання у малопотужних пристроях. Системи з «energy harvesting» можуть використовуватися у моніторах тиску в автомобільних шинах, бездротових метеостанціях, імплантованих медичних пристроях, дорожніх попереджувальних знаках, марсоходах, вузлах WSN (бездротова сенсорна мережа), пристроях IoT (Інтернет речей) тощо.

Видів «energy harvesting» велика кількість, але кожний розрізняється за деякими параметрами. Нижче наведена загальна таблиця видів «energy harvesting».

Пристрої, що працюють за технологією «energy harvesting» для отримання енергії, здебільшого, реалізовані на основі MEMS. Вони використовують вібрації навколишнього середовища для перетворення їх в корисну енергію.

На рисунку 1.2 зображена загальна модель сенсорної системи з використанням «energy harvesting» в якості альтернативного джерела енергії. Енергія отримана з блоку «energy harvesting» спочатку потрапляє до блоку, що аналізує та передає енергію до вбудованої системи у мікроконтролері.

Модель сенсорної системи, яка наведена на рисунку 1.2 використовує алгоритм оптимізації енергоспоживання для живлення мікроконтролера, датчиків та модулів системи. Такий варіант надає можливість використовувати у вбудованій системі повний обсяг отриманої енергії за технологією «energy harvesting».

Таблиця 1.1 – Загальна таблиця видів «energy harvesting»

Тип	Переваги	Недоліки
Електростатичний	Не потребує смарт матеріалу Масштабований Напруга 2-10 В	Зовнішнє джерело напруги Необхідні механічні обмеження
П'єзоелектричний	Не потребує зовнішнього джерела напруги Висока напруга > 10 В Сумісний з MEMS	Деполаризація Високий вихідний опір Витік заряду
Магнітострикційний	Надвисокий коефіцієнт зв'язку > 0,9 Відсутність проблеми з деполаризацією Високочастотна вібрація	Нелінійний ефект Знімна котушка Масштабованість
Електромагнітний	Немає розумного матеріалу Немає зовнішнього джерела напруги	Масштабованість Низька напруга на виході

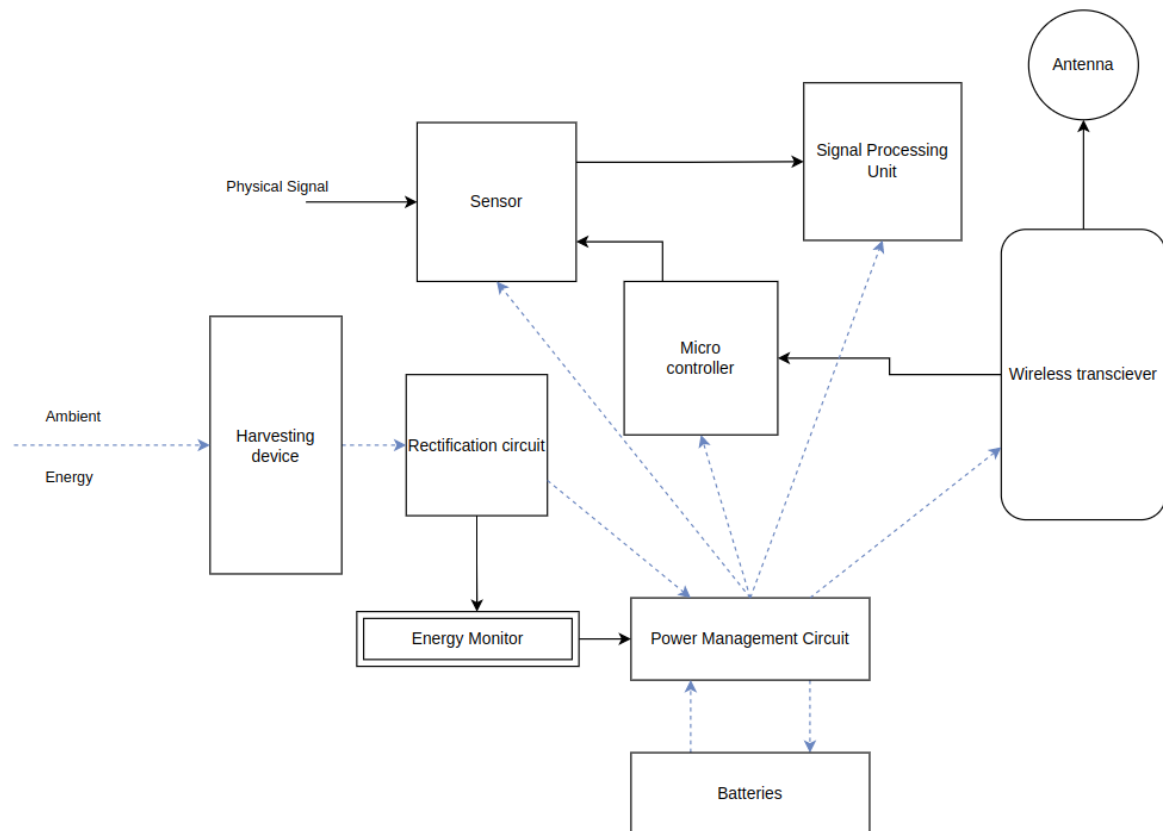


Рисунок 1.2 – Модель сенсорної системи з використанням «energy harvesting» в якості альтернативного джерела енергії

1.3 Методи динамічного керування живленням

На теперішній час існують методи динамічного керування живленням для використання технології «energy harvesting».

Методи динамічного керування живленням (DPM) вибірково переводять компоненти системи в стани низького енергоспоживання, коли вони неактивні. Систему керуваного живлення можна змоделювати як машину стану живлення, де кожен стан характеризується енергоспоживанням і продуктивністю. Крім того, переходи між станами мають витрати на потужність і затримку. Зазвичай менше енергоспоживання також означає нижчу продуктивність і більшу затримку переходу. Коли компонент переходить у стан низького енергоспоживання, наприклад у стан сну, він недоступний протягом періоду часу, проведеного там, а також протягом часу переходу між станами. Переходи

між станами контролюються командами, виданими диспетчером живлення (PM), який спостерігає за робочим навантаженням системи та вирішує, коли та як примусово змінювати стан живлення. Менеджер живлення приймає рішення про перехід стану відповідно до політики керування живленням.

Зменшення потужності також здійснюється за допомогою алгоритмів динамічного масштабування напруги (DVS). Замість довшого періоду простою центральний процесор (ЦП) сповільнюється до точки, коли він виконує задачу вчасно для надходження наступного запиту на обробку, водночас економлячи досить багато енергії. Алгоритми DVS зазвичай реалізуються на рівні планувальника операційної системи (ОС).

Нарешті, багато останніх робіт розглядали об'єднання DVS і DPM в єдину реалізацію керування живленням. Коротші періоди простою краще піддаються DVS, тоді як триваліші більш підходять для DPM. Таким чином, для отримання оптимальних результатів необхідна комбінація двох підходів.

Більшість комерційних реалізацій керування живленням зосереджені лише на вирішенні, коли пристрій повинен перейти в режим сну. Вартість переходу між активним і сплячим станом зазвичай визначає, наскільки жорсткий алгоритм використовується в системі. Якщо необхідність досить низька або, принаймні, не сприймається користувачами, тоді алгоритм переводить пристрій у режим сну, щойно він стає неактивним. Наприклад, «μSleep» було представлено як спосіб зменшення енергоспоживання в режимі очікування платформи HP IPAQ [41]. Наручний годинник IBM переходить у режим очікування, як тільки він стає неактивним [42]. Технологія Intel QuickStart переводить процесор у режим сну між натисканнями клавіш. У всіх трьох випадках вартість пробудження не сприймається користувачем. Коли вартість переходу до стану низького енергоспоживання є значною, комерційні програми зазвичай реалізують алгоритми на основі часу очікування.

Існуючі алгоритми прогнозування намагаються не лише передбачити тривалість наступного часу простою шляхом вивчення розподілу надходжень запитів, але й зробити це з достатньою точністю, щоб можна було перевести пристрій у стан сну без простою. Крім того, деякі алгоритми прогнозування

також виводять пристрій із сплячого стану в очікуванні надходження запиту на обслуговування. Якщо передбачені терміни та тривалість періоду простою правильні, тоді прогнозні алгоритми забезпечують рішення без додаткових витрат. З іншого боку, якщо прогноз помилковий, потенційні витрати можуть бути досить великими.

1.4 Постановка задачі

На основі проведеного аналізу було сформульовано мету та задачі, які необхідно вирішити для реалізації системи спостереження за станом здоров'я людини з використанням технології «energy harvesting».

Метою роботи є розробка системи спостереження за станом здоров'я людини з використанням, в якості джерела енергії «energy harvesting», із застосуванням алгоритму оптимізації енергоспоживання системи.

Для досягнення поставленої мети повинні бути вирішені наступні завдання:

- дослідження існуючих алгоритмів оптимізації енергоспоживання системи та вибір підходящого для реалізованої моделі;
- розробка загальної моделі системи відстеження показників здоров'я з використанням, в якості джерела енергії «energy harvesting».
- вибір мікроконтролеру;
- вибір модуля п'єзоелектричного перетворювача енергії для використання технології «energy harvesting»
- вибір датчиків та сенсорів системи;
- аналіз даних про енергозатратність системи мікроконтролера та окремих елементів;
- аналіз можливості переведення окремих компонентів системи в режим сну;
- розробка системи для спостереження за станом здоров'я;

- написання програмного забезпечення для роботи системи спостереження;
- аналіз необхідності використання алгоритму енергозбереження;
- реалізація алгоритму енергоефективності системи;
- створення тестів для перевірки правильної роботи програми.

2 РОЗРОБКА СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА СТАНОМ ЗДОРОВ'Я ЛЮДИНИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ENERGY HARVESTING

2.1 Вибір мікроконтролеру

Для реалізації системи спостереження за станом здоров'я людини в якості основного модулю управління було обрано плату Seeeduno Mega на основі AVR мікроконтролера. Мікроконтролери AVR мають гарвардську архітектуру (програма та дані знаходяться у різних адресних просторах) та систему команд, близьку до ідеології RISC. Процесор AVR має 32 8-бітових регістра загального призначення, об'єднаних у регістровий файл. На відміну від «ідеального» RISC, регістри не є абсолютно ортогональними:

- деякі команди працюють лише з регістрами r16...r31. До них належать команди, які працюють з безпосереднім операндом: ANDI/CBR, ORI/SBR, CPI, LDI, LDS (16-біт), STS (16-біт), SUBI, SBCI, а також SER та MULS;
- команди, які збільшують і зменшують 16-бітове значення (у тих моделях, де вони доступні) з безпосереднім операндом (ADIW, SBIW), працюють тільки з однією з пар r25:r24, r27:r26(X), r29:r28(Y) або R31: R30 (Z);
- команда копіювання пари регістрів (у тих моделях, де доступна) працює тільки з сусідніми регістрами, що починаються з непарного (r1: r0, r3: r2, ..., r31: r30);
- результат множення (у тих моделях, у яких є модуль множення) завжди міститься в r1: r0. Також тільки ця пара використовується як операнд для команди самопрограмування (де доступна);
- деякі варіанти команд множення приймають як аргументи лише регістри з діапазону r16...r23 (FMUL, FMULS, FMULSU, MULSU).

Обрана плата Seeeduino Mega, що заснована на мікроконтролері ATmega 2560, повністю сумісна з платою Arduino Mega. А також, з усіма платами розширення, розробленими для платформ Uno, Duemilanove або Diecimila. Розташування пінів 0 - 13, аналогових входів 0 - 5, силового роз'єму, блоку ICSP, порту послідовної передачі UART (виходи 0 і 1) та зовнішнього переривання 0 і 1 на Mega відповідає розташуванню на вищенаведених платформах.

Seeeduino Mega має 70 контактів цифрового входу/виходу (з яких 14 можна використовувати як виходи ШІМ), 16 аналогових входів, 4 UART (апаратних послідовних портів), кварцевий генератор на 16 МГц, з'єднання по USB, роз'єм живлення, роз'єм ICSP і кнопка налаштування. Кожен із 70 цифрових контактів на Mega можна використовувати як вхід або вихід за допомогою функцій `pinMode()`, `digitalWrite()` і `digitalRead()`. Вони працюють при напрузі 5 та 3.3 вольт.

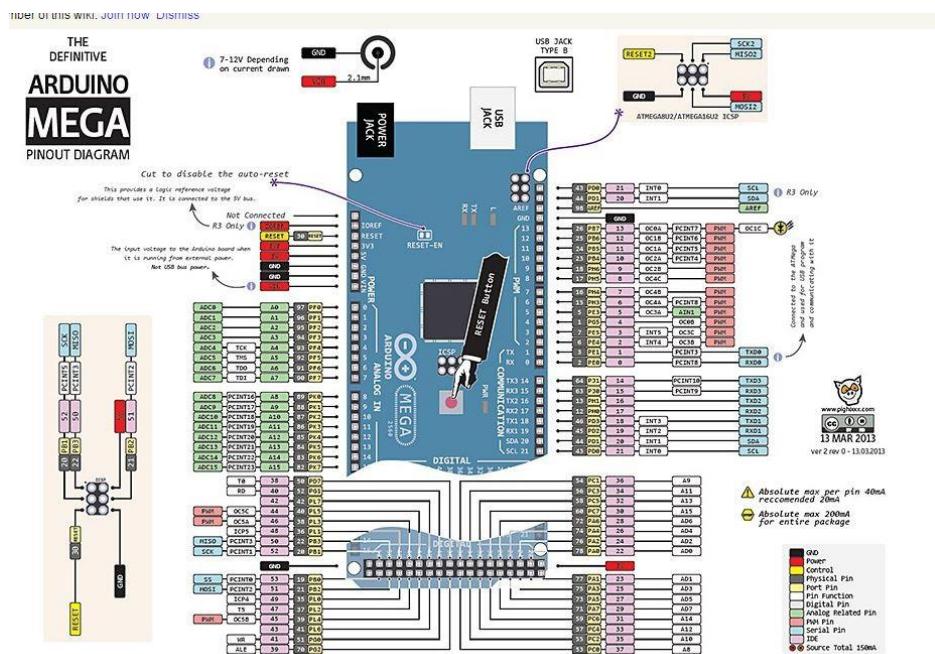


Рисунок 2.1 – Принципова схема Arduino Mega 2560

Плата містить усе необхідне для розробки системи, а також елементну об'язку для програмування та інтеграції з іншими схемами. На кожній платі

обов'язково присутні: лінійний стабілізатор напруги 3.3 та 5 В і 16 МГц кварцевий генератор (в деяких версіях керамічний резонатор). На рисунку 2.1 можна побачити принципову схему Arduino Mega 2560.

Плата Seeeduno Mega базується на мікроконтролері ATmega 2560. За даними цього мікроконтролера, отриманими з документації, можна зробити висновки щодо енергоспоживання системи, так як це одна з найпотужніших частин приладу. Отримані дані наведені в таблиці 2.1.

Таблиця 2.1 – Дані про кількість енергії для живлення ATmega 2560.

Режим	Умова	Типічне значення	Максимальне значення	Одн
Активний режим	Active 1MHz, Vcc = 2V (ATmega640/1280/2560/1V)	0.5	0.8	мА
	Active 4MHz, Vcc = 3V (ATmega640/1280/2560/1L)	3.2	5	
	Active 8MHz, Vcc = 5V (ATmega640/1280/1281/2560/2561)	10	14	
	Idle 1MHz, Vcc = 2V (ATmega640/1280/2560/1V)	0.14	0.22	
	Idle 4MHz, Vcc = 3V (ATmega640/1280/2560/1L)	0.7	1.1	
	Idle 8MHz, Vcc = 5V (ATmega640/1280/1281/2560/2561)	2.7	4	
Режим сну	WDT enabled, Vcc = 3V	До 5	15	мкА
	WDT disabled, Vcc = 3V	До 1	7.5	

Таблиця 2.2 – Дані про кількість енергії для живлення LM35

Режим	Варіанти тестування		LM35CA		Одн.
			Типічні показники	Максимальні показники	
Активний режим	$V_s = 5 \text{ V}, 25^\circ\text{C}$	Tested Limit	56		мкА
		Design Limit		67	
	$V_s = 5 \text{ V}, -40^\circ\text{C} \leq T_j \leq 125^\circ\text{C}$	Tested Limit	105		
		Design Limit		131	
	$V_s = 30 \text{ V}, 25^\circ\text{C}$	Tested Limit	56.2		
		Design Limit		68	
$V_s = 30 \text{ V}, -40^\circ\text{C} \leq T_j \leq 125^\circ\text{C}$	Tested Limit	105.5			
	Design Limit		133		
Режим сну	$4 \text{ V} \leq V_s \leq 30 \text{ V}, 25^\circ\text{C}$	Tested Limit	0.2		
		Design Limit		1	
	$4 \text{ V} \leq V_s \leq 30 \text{ V}, -40^\circ\text{C} \leq T_j \leq 125^\circ\text{C}$	Tested Limit	0.5		
		Design Limit		2	

Згідно з таблицею даних про кількість енергії для живлення датчику температури LM35, можна зробити висновки, що кількість необхідної енергії при максимальному завантаженні буде дорівнювати близько 0,1мА.

Температуру тіла пацієнта було запропоновано вимірювати за допомогою MLX90614. MLX90614 – це інфрачервоний термометр для безконтактного вимірювання абсолютної температури об'єктів. Мікросхема термодетектора є чутливою до інфрачервоного випромінювання.

До MLX90614 інтегровано підсилювач з низьким рівнем шуму, 17-розрядний АЦП і потужний блок DSP, що забезпечує високу точність термометра.

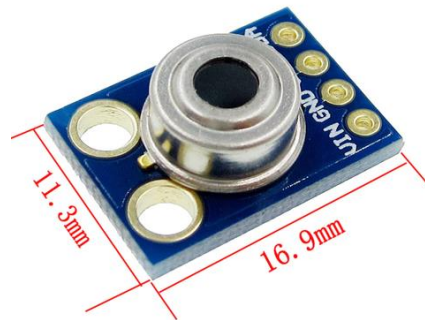


Рисунок 2.3 – Датчик температури MLX90614

Таблиця 2.3 – Дані про кількість енергії для живлення MLX90614.

Параметри	MLX90614ESF- Ахх	MLX90614ESF- Вхх	MLX90614KSF- Ахх
DC sink current, SDA / PWM pin	25 mA		
DC source current, SDA / PWM pin	25 mA		
DC clamp current, SDA / PWM pin	25 mA		
DC clamp current, SCL pin	25 mA		

Датчик MLX90614 потребує більшого споживання енергії, ніж датчик температури LM35. Він потребує близько 25 мА для використання цифрового входу/виходу. Де, у нормальному режимі, зберігається виміряна температура об'єкта. При цьому, переключення до режиму сну майже не впливає на результат використання енергії, тому не є раціональним, вимикати даний датчик.

Датчик електричної активності серця AD8232, розроблений компанією Analog Devices, на 50% компактніший і використовує на 20 відсотків менше

енергії, ніж аналогічні пристрої. Він складається з плати AD8232, набору електродів та дротів для їх підключення до плати AD8232. Датчик призначений для отримання, посилення та фільтрації біопотенційних сигналів, навіть за умов сильних перешкод. AD8232 складається з двополюсного фільтру високих частот і операційного підсилювача, який дозволяє використовувати технологію багатополюсної низькочастотної фільтрації для видалення шуму лінії та інших перешкод. Завдяки цьому датчик можна підключати безпосередньо до осцилографу.



Рисунок 2.4 – Модуль вимірювання імпульсів електричної активності серця AD8232

Останій датчик системи AD8232 має більш однозначні дані щодо витрачання енергії. Йому необхідно всього 170 мкА для роботи. Це невеликий обсяг, тому немає необхідності переключати його в режим сну.

Отримані дані було вирішено виводити у додаток, що встановлений на смартфоні користувача. У якості модуля зв'язку був обраний мікроконтролер ESP8266 Serial WIFI.

Модуль ESP8266 являє собою не тільки бездротовий приймач-передавач по стандарту Wi-Fi, а й повноцінний мікроконтролер. Можна скористатися вже вбудованою прошивкою і за допомогою інтерфейсу SDIO 2.0, SPI, UART вести передачу даних. ESP8266 має вбудовані інтерфейси Wi-Fi + BT + BLE. Він може використовуватися в мережевих пристроях обробки даних та обслуговування датчиків.



Рисунок 2.5 – Модуль ESP8266 Serial WIFI

Характеристики споживання енергії ESP8266 розділяються на два основних режими: приймання та передача даних. В режимі приймання даних, рівні споживання струму сильно залежать від активного перетворювача напруги та швидкості обміну даними. Додаткова енергія витрачається на запуск та налаштування модулю перед початком безпосереднього отримання даних. Таким чином споживання енергії у режимі прослуховування можна знизити за рахунок використання DC-DC перетворювача замість LDO та зменшення швидкості обміну даними. Дані про ESP8266 наведено в таблиці 2.4.

Таблиця 2.4 – Дані про кількість енергії для живлення ESP8266

Parameres	Min	Typical	Max	Unit
TX802.11b, CCK 11 Mbps, Pout = +17 dBm	-	170	-	mA
TX802.11 g, OFDM 54Mbps, Pout = +15 dBm	-	140	-	
TX802.11 n, MCS7, Pout = +13 dBm	-	120	-	
Rx802.11 b, 1024 bytes packet length, - 80 dBm	-	50	-	
Rx802.11 b, 1024 bytes packet length, - 70 dBm	-	56	-	
Rx802.11 b, 1024 bytes packet length, -65dBm	-	56	-	

Отримані дані свідчать про те, що робота передачі даних по мережі є доволі енерговитратною. Однак, технологія 802.11n пропонує найбільш економний варіант використання мережі. У режимі очікування система витрачає 12 мА, а при надсиланні повідомлень витрачає від 120 мА до 170 мА. Можна зробити висновки, що найкращим вибором буде використання приймача, що підтримує технологію 802.11n.

Окрім того, ESP8266 має режими енергозбереження. За винятком стандартних режимів повного функціонування та вимкненого стану, модуль WeMos NodeMCU Lua підтримує енергозберігаючі режими, призначені для кожного окремого випадку. Існуючі режими енергозбереження описані нижче:

- режим Modem-sleep використовується у випадках, коли WiFi-зв'язок встановлений та постійна передача даних не потрібна, при цьому схема WiFi-модему може бути відключена для економії енергії. Наприклад, у режимі DTIM3, коли ESP8266 "спить" 300 мілісекунд і прокидається на 3 мілісекунди для прийому від точки доступу пакетів бездротових маяків (Beacon), загальне споживання струму становить близько 20мА;

- режим Light-sleep використовується в задачах, в яких підтримка з'єднання WiFi і передачі даних не потрібна, при цьому процесор може бути призупинений. Наприклад, режим комутатора WiFi. Загальне середнє споживання струму становить близько 2 мА;

- режим Deep-sleep – режим глибокий сон, що є оптимальним для додатків, які не вимагають підключення WiFi і передають дані з більшими затримками часу. До таких завдань відносяться датчики температури, що виконують вимірювання кожні 100 секунд. Наприклад, коли ESP8266 спить 300 секунд і прокидається для з'єднання з точкою доступу (близько 0.3-1 секунди), загальне середнє споживання струму набагато менше 1 мА.

2.3 Вибір джерела живлення системи

В якості джерела енергії використовується пристрій отримання енергії з реалізацією технології «energy harvesting». Багато таких пристроїв розроблені

провідними компаніями світу. Існує достатня кількість варіацій, для використання у різних умовах, та для потреб різних систем. На даний момент одними з популярних пристроїв отримання енергії є модулі п'єзоелектричного перетворювача енергії LTC3588-1 та LTC3330. Вони належать до RE01 MCU родини та можуть видавати близько 100 мА струму. Вихідний рівень напруги встановлюється на два контакти і має чотири можливих значення: 1.8, 2.5, 3.3 і 3.6 В, що підходить для живлення мікроконтролерів, датчиків і бездротових приймачів.

Енергії від системи «energy harvesting» повинно вистачати на повну функціональність малопотужної системи. Для повноцінної роботи, здебільшого, використовується блок контролю зарядом. Він необхідний для регулювання необхідної кількості заряду, що подається на мікроконтролер, або перенапрявлення заряду для накопичення в акумуляторі системи.

В якості джерела живлення в реалізованій системі використовується модуль LTC3588-1, що базується на використанні технології «energy harvesting». Даний модуль складається з двоперіодного мостового випрямляча із низькими витратами та високоефективного понижуючого перетворювача. Він є готовим рішенням для збору енергії, оптимізованим під джерела енергії з високим вихідним опором, такі як п'єзоелектричні, сонячні або магнітні перетворювання. LTC3588-1 працює при вхідній напрузі до 20 В. При перевищенні цього значення спрацьовує захисний шунт. П'єзоелемент підключається до виводів PZ1 та PZ2 на LTC3588-1, з'єднаних із вбудованим мостовим випрямлячем. А вже, за допомогою випрямленої напруги заряджається вхідний конденсатор.

Між LTC3588-1 та вбудованою системою розташований блок, який відповідає за кількість енергії, що поступає у вбудовану систему, а надлишки надає до блоку живлення аби використовує в періоди відсутності енергії, необхідної для роботи вбудованої системи.

Модуль LTC3588-1 надає можливість системі не бути прив'язаною до стаціонарного джерела енергії. Даний модуль призначений для безпосереднього підключення до п'єзоелектричного джерела живлення змінного струму,

випрямлення форми сигналу напруги та зберігання зібраної енергії на зовнішньому конденсаторі, відведення будь-якої надлишкової потужності через внутрішній регулятор і підтримки регульованої вихідної напруги за допомогою нанопотужного високоефективного синхронного понижувального регулятора.



Рисунок 2.6 – Модуль LTC3588-1

В даному модулі використовується режим блокування низької напруги наднизького струму спокою (UVLO), що дозволяє заряду накопичуватися на вхідному конденсаторі, поки понижуючий перетворювач не зможе ефективно передавати частину накопиченого заряду на вихід. При відсутності енергії в оточуючому середовищі, LTC3588-1 переходить у стан сну, у якому як вхідний, так і вихідний струми спокою мінімальні. На рисунку 2.7 представлена принципова схема LTC3588-1.

Збільшення ємності вихідного аккумулятора надає можливість накопичувати необхідну для пристроя кількість енергії. Але все одно, необхідно оптимізувати витрачання енергії у вбудованій системі. Для цього необхідно розробити програму, яка зможе стежити за ступінем розрядженості пристрою та підвищувати чи понижати енергоспоживання системи без втрати її ефективності.

100mA Piezoelectric Energy Harvesting Power Supply

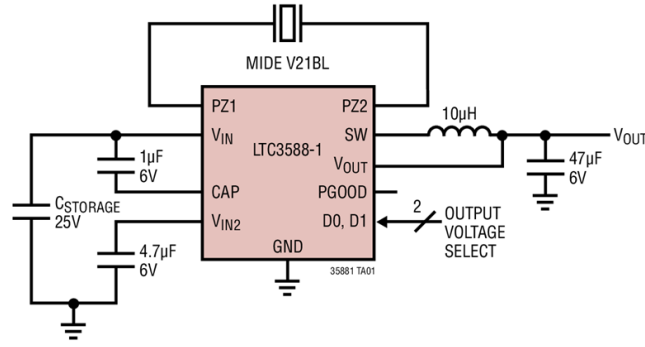


Рисунок 2.7 – Принципова схема LTC3588-1

2.4 Реалізація моделі

Модель реалізованої системи та її живлення зображена на рисунку 2.8 та показує всі елементи системи та їхній взаємозв'язок. Вхідна енергія проходить через блок «energy harvesting», блок керування живленням, а вже потім отримується мікроконтролером, а від нього поступає до датчиків та Wi-Fi - модуля.

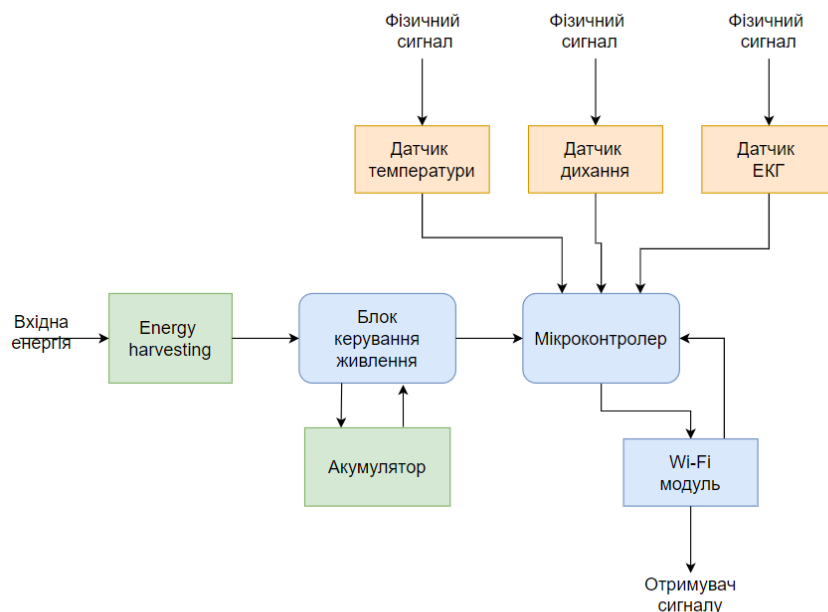


Рисунок 2.8 – Схема реалізованої моделі спостереження за станом здоров'я людини

2.5 Енергоспоживання приладу

Енергоспоживання готового приладу можна оцінити як суму енергії, що спожита кожним з елементів системи:

$$E_{\text{total}} = \sum_{i=1}^n E_i, \quad (2.1)$$

де E_{total} – загальна кількість спожитої енергії, а E_i – індивідуальний показник спожитої енергії для кожного з елементів сенсорного вузла.

За формулою (2.1) та отриманих з документацій характеристик елементів, можна розрахувати необхідні дані. А саме, максимальне та мінімальне значення необхідного струму для системи. У таблиці 2.5 наведені отримані значення енергоспоживання елементів у режимі очікування та максимальне значення в режимі роботи.

Таблиця 2.5 – Загальні значення системи

Елемент	Режим очікування	Максимальне значення
ATmega 2560	15 мкА	14 мА
LM35 x 2	0,4мкА	0,2мА
MLX90614	25 мА	
AD8232	170 мкА	
ESP8266	1 - 20 мА	170 мА
Всього	46,7мА	209,3мА

Згідно з отриманими результатами та обчисленнями можна зробити висновки, щодо засобів запобігання ситуацій вимкнення пристрою від недостачі живлення. У розробленій системі необхідно реалізувати такі можливості запобігання відключення системи:

- по-перше, застерігати користувача, про недостачу струму та відправляти цю інформацію на РК-дісплей та на смартфон;
- по-друге, в даному пристрої повинне бути реалізовано управління циклами сну/роботи для перешкодження ситуацій закінчення заряду. Даний алгоритм повинен враховувати особливості роботи усіх компонентів системи, необхідність у їхній постійній роботі, та раціональності їх відключення.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СПОСТЕРЕЖЕННЯ ЗА СТАНОМ ЗДОРОВ'Я

3.1 Компоненти реалізованої системи

У відповідності до структури, описаної у розділі 2.4 даної записки, було розроблено тестову модель системи спостереження за станом здоров'я. Схема реалізованої системи зображена на рисунку 2.1. До системи входять:

- плата Seeeduino Mega, що базується на мікроконтролері ATmega2560;
- джерело живлення (модуль LTC3588-1);
- датчики температури та ЕКГ.

3.2 Методи оптимізації енергоефективності системи

За результатами дослідження з розділу 2, було прийнято рішення реалізувати алгоритм управління циклами сну/роботи для системи моніторингу стану здоров'я людини з використанням технології «energy harvesting» в якості джерела енергії.

Згідно з результатами дослідження були зроблені такі висновки стосовно кожного з елементів системи:

- мікроконтролер ATmega 2560. Для економії енергії на даному мікроконтролеру пропонується використати один з алгоритмів оптимізації, таких як DVS і DPM, або їх поєднання;
- LM35 – даний датчик було прийнято залишати в режимі роботи. Значних змін при переході до режиму сну не відбувається. Тому не має сенсу вмикати та вимикати цей датчик;

- цифровий датчик температури MLX90614 та датчик електричної активності серця AD8232. Ці датчики не змінюють, або майже не змінюють свої показники в режимі сну, тому вирішено залишити їх працювати постійно, без відключень. Крім того, дані з цих датчиків є важливими для аналізу стану людини та подальшої роботи системи;

- ESP8266 в якості модуля Wi-Fi. Одним з важливих факторів є те, що ESP8266 може використовувати технологію 802.11n, а це означає, що при найбільшій з доступних ESP8266 швидкостей можливо передавати дані при мінімальних енерговитратах. Крім того, за допомогою режиму Deep-sleep можна витратити лише 1mA при цьому відправляючи дані на мобільний пристрій раз в секунду, чого буде достатньо для отримання актуальних даних.

3.3 Огляд мови програмування розробки

Реалізація даного алгоритму була виконана на мові програмування C/C++. Вона була обрана за декількома параметрами. Насамперед це швидкодія мови. Адже даний алгоритм розроблений для того, щоб покращити енергоефективність та швидкодію приладів. Крім того, використання C/C++ дозволяє працювати безпосередньо з пам'яттю, адресами, портами на низькому рівні. Тому дана мова програмування є виїгральною в даному питанні. Мова програмування пристроїв Arduino ґрунтується саме на C/C++. Він простий у освоєнні, і зараз Arduino - це, один з найзручніших способів програмування пристроїв на мікроконтролерах ATmega2560.

3.4 Огляд середовища розробки - Arduino IDE

Інтегроване середовище розробки – це комплексне програмне рішення для розробки програмного забезпечення. Типове середовище розробки містить:

- редактор програмного коду;
- компілятор та/або інтерпретатор;

- засоби автоматизації збірки продукту;
- відладчик.

Для розробки демонстраційного програмного забезпечення для плати було обрано інтегроване середовище розробки Arduino IDE (рис. 3.1).

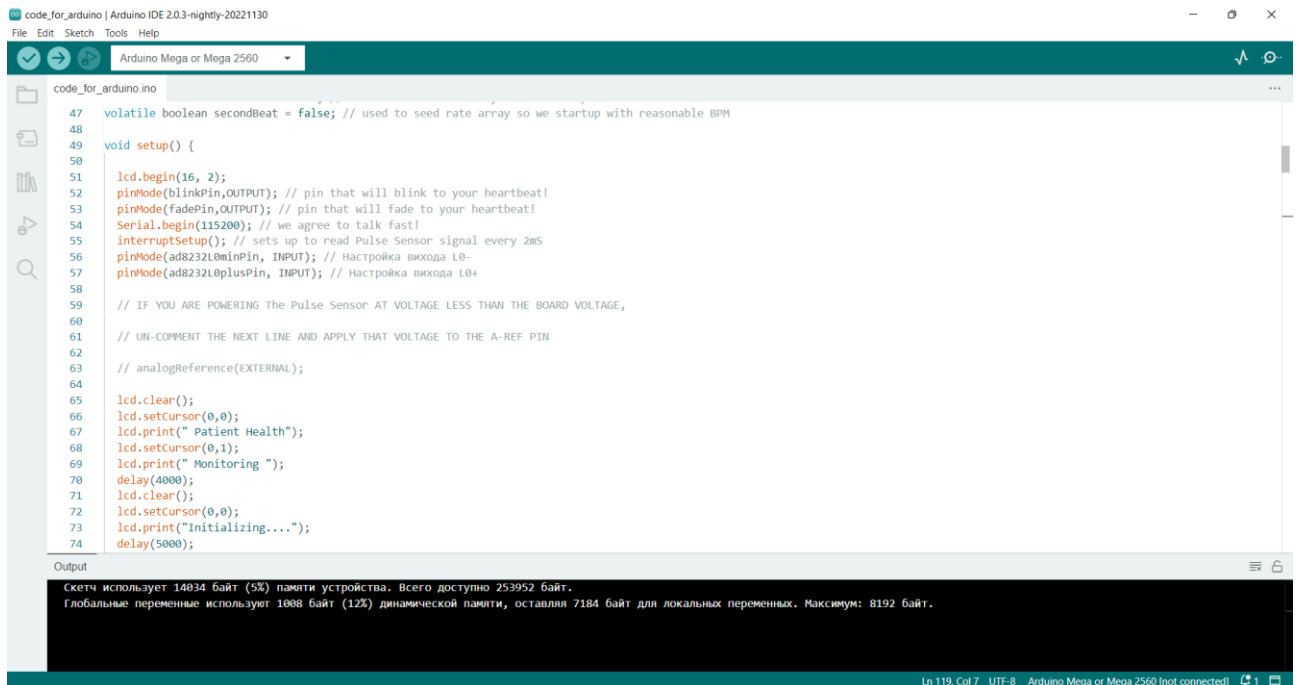


Рисунок 3.1 – Інтегроване середовище розробки Arduino IDE

Arduino Integrated Development Environment, або програмне забезпечення Arduino (IDE), містить текстовий редактор для написання коду, область повідомлень, текстову консоль, панель інструментів із кнопками для функцій та меню.

Програма, написана в середовищі розробки Arduino, називається скетч. Скетч пишеться в текстовому редакторі, який має інструменти вирізки/вставки, пошуку/заміни тексту. Під час збереження та експорту проекту в області повідомлень з'являються пояснення, також можуть відображатися помилки. Вікно виведення тексту (консоль) показує повідомлення Arduino, які містять повні звіти про помилки та іншу інформацію.

Вихідний код середовища випущений під загальнодоступною ліцензією GNU версії 2. Підтримує мови Сі та С++ з використанням спеціальних правил структурування коду. Arduino IDE використовує бібліотеку програмного забезпечення Wiring, яка надає безліч загальних процедур.

Wiring — це платформа програмування мікроконтролерів з відкритим кодом. Вона дозволяє писати кросплатформне програмне забезпечення для керування пристроями, підключеними до широкого спектру плат мікроконтролерів, для кодування та роботи з інтерактивними об'єктами.

З використанням цієї бібліотеки відбувається проста інтеграція апаратного забезпечення від виробника Atmel. Поточне ядро AVR8 підтримує апаратне забезпечення Wiring і будь-яке обладнання на основі процесорів AVR ATmega.

3.5 Структура програмної частини

З того, що було проаналізовано в пункті 2.5, та обчислено у таблиці 2.5, можна зробити висновки, що в режимі очікування система втрачає набагато менше енергії, ніж при використанні усіх можливих функцій на максимумі. Окрім того, що останній режим майже неможливо використовувати, особливо на постійній основі. Але було прийнято рішення реалізувати алгоритм, що буде відстежувати необхідність кожного окремого компонента та системи в цілому та регулювати її енерговитрати.

З отриманих даних зрозуміло, що витрачання енергії на мінімумі при режимах сну дорівнює близько 47мА, на відміну від використання на максимум, що досягає 209мА. Виходить досить помітна різниця, яку необхідно використати на користь.

Реалізований у роботі алгоритм припускає переключання деяких з компонентів на невеликий час у режим сну. Відключення окремих компонентів, а час від часу і системи повністю, призводить до сильного зменшення енергоспоживання системи. Таке регулювання не зробить нічого поганого,

адже відключення на 30-60 секунд не є критичним у даному випадку.

При цьому реалізована модель постійно відстежує наявність екстрених симптомів, про які потрібно повідомити негайно. Після кожного виміру усіх даних (температура, електрокардіографія) на платі Seeeduino Mega відбувається аналіз, порівняння з контрольною групою пацієнта, та у випадку необхідності посилається екстрений сигнал на мобільний пристрій, прив'язаний до пристрою.

Блок-схема реалізованого алгоритму пропонується на рисунку 3.3. На ній зображені основні етапи процесу відключення системи від активного енергоспоживання та перехід у режим сну.

Запропонована в роботі система заснована на визначенні необхідного набору функцій для користувача, залежності кожного елемента від кількості струму та доцільності переключення системи та кожного з компонентів в режим сну.

Основними етапами виконання алгоритма є дані функції: `cmpEnergy()`, `cmpData()`, `emergencyCall()`, `sleepMode(param)`, `getData(param)`, `sendData(param)`, `sleepModeAll()` та `timer()`. Кожна з них може включати в себе багато інших функцій, але основна реалізація виконується саме в них. Далі буде розглянуто основні функції кожної.

Функція `cmpEnergy()` порівнює та аналізує кількість необхідної енергії для системи в даний момент. Порівнює ці дані з отриманими від блоку керування енергією, що вбудований у модуль «energy harvesting». Та подає подальшу інформацію, щодо виконання алгоритму. Після виконання даної функції розробляється подальший план роботи. Система або починає підготовку до переходу в режим сну, або продовжує працювати в нормальному режимі.

Функція `cmpData()` порівнює як давно в останій раз були отримані результати від датчиків, та відправлені по мережі. При очікуванні, що скоро повинні прийти нові результати, відпраляє сигнал до наступної функції, про неготовність елемента до переходу в режим сну. Або при готовності переходить до функції `sleepModeAll()`. Очікується, що даний алгоритм подій,

буде використовуватися рідко, тому що, не так часто система може відключитися одразу, без того, щоб завершити незакінчену роботу.

Функція `emergencyCall()` необхідна для того, щоб обчислювати дані, що прийшли від датчиків, аналізувати їх, порівнюючи з контрольною групою пацієнта та викликати термінову відправку повідомлення по мережі Wi-Fi у разі необхідності.

Функція `sleepMode(param)` відключає лише ті елементи моделі, які вже закінчили виконання своїх задач та можуть перейти до режиму сну. Які це саме параметри, розраховується у функції `cmpData()`.

Функція `getData(param)` отримує параметри від елементів, що не завершили цикл роботи. Це необхідно для того, щоб відправити дані від датчиків на приймаючий пристрій, коли вони закінчують поточний цикл, та перевести їх до режиму сну.

Функція `sendData(param)` виконує відправку усіх та окремих параметрів по мережі Wi-Fi. Ця ж функція використовується при виклику функції `emergencyCall()`, але терміново.

Функція `sleepModeAll()` переводить у режим сну одразу усі компоненти системи без очікування на погодження або отримання даних від компонентів моделі. Очікується, що дана функція буде використовуватися рідко, і в періоди критичної нестачі електроенергії.

Функція `timer()` необхідна для того, щоб система не відключалася та вмикалась постійно. Розраховано, що приблизно 30 секунд система буде знаходитися у стані сну, окрім окремих елементів, що повинні слідкувати за станом пацієнта постійно.

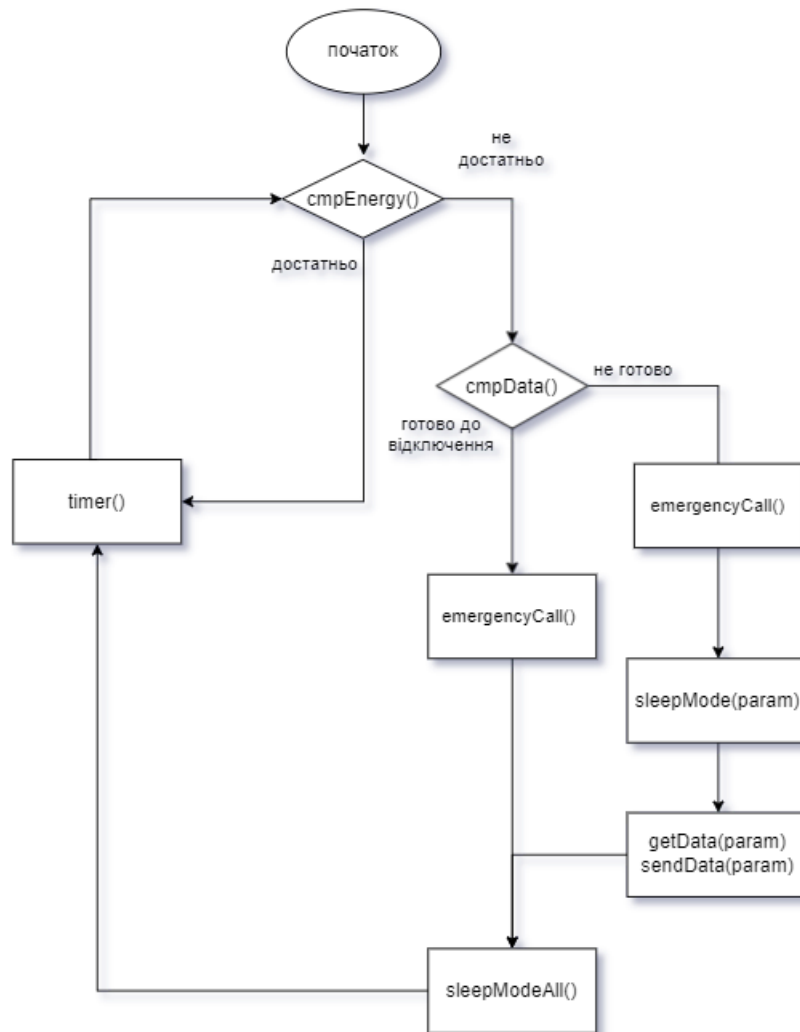


Рисунок 3.2 – Блок-схема реалізованого алгоритму

3.6 Використані бібліотеки

У ході написання програми були використанні такі бібліотеки:

– SoftwareSerial.h. Бібліотека SoftwareSerial дозволяє реалізувати послідовний інтерфейс на будь-яких інших цифрових виводах Arduino за допомогою програмних засобів, що дублюють функціональність UART (звідси і назва SoftwareSerial). Бібліотека дозволяє програмно створювати кілька послідовних портів, що працюють на швидкості до 115 200 біт. Для пристроїв,

що працюють з інвертованим сигналом, у бібліотеці передбачено відповідний параметр, що включає інвертування;

– LiquidCrystal.h. Ця бібліотека дозволяє Ардуїно керувати рідкокристалічними дисплеями (LCD), побудованими на базі поширеного чіпсету Hitachi HD44780. У бібліотеці реалізовано як 4-х, так і 8-бітовий режими роботи (тобто є можливість використовувати 4 або 8 ліній даних, спільно з керуючими лініями RS, Enable та RW);

– Adafruit_MLX90614.h. Бібліотека, що дозволяє користуватися датчиком температури MLX90614;

– Wire.h. Ця бібліотека дозволяє Ардуїно взаємодіяти з різними пристроями через інтерфейс I2C/TWI. На платах Ардуїно версії R3 (з розпинуванням 1.0) лінії SDA (дані) та SCL (тактові імпульси), пов'язані з цим інтерфейсом, розташовані на роз'ємах біля контакту AREF. В Arduino Due реалізовано два інтерфейси I2C/TWI, лінії одного з них (SDA1 та SCL1) розташовані біля виводу AREF, лінії другого – на виводах 20 та 21;

– avr/wdt.h. Цей файл заголовка оголошує інтерфейс до деяких вбудованих макросів, що обробляють сторожовий таймер, присутній у багатьох пристроях AVR. Щоб запобігти випадковій зміні конфігурації сторожового таймера програмою, що виходить з ладу, для її зміни потрібна спеціальна послідовність часу. Макроси в цьому файлі заголовка автоматично обробляють необхідну послідовність перед зміною будь-якого значення. Під час маніпуляції переривання будуть вимкнені;

– avr/sleep.h. Використання інструкції SLEEP може дозволити програмі значно зменшити енергоспоживання. Пристрої AVR можна переводити в різні режими сну. У цьому файлі заголовка є кілька макросів, які фактично переводять пристрій у сплячий режим. Один зі способів - встановити потрібний режим сну за допомогою set_sleep_mode() (зазвичай за замовчуванням це режим очікування, коли ЦП переводиться в режим сну, але всі периферійні годинники все ще працюють), а потім викликати sleep_mode(). Цей макрос

автоматично встановлює біт увімкнення сну, переходить у режим сну та очищає біт увімкнення сну. Є п'ять режимів сну на вибір: режим очікування, вимкнення живлення, збереження сили, розширений режим очікування. Щоб увійти в будь-який із режимів сну, біт увімкнення режиму сну в регістрі керування режимом сну (SMCR.SE) має бути записаний у значення «1» і повинна бути виконана інструкція SLEEP. Біти вибору режиму сну (SMCR.SM[2:0]) визначають, який режим сну (неактивний, вимкнення, енергозбереження, очікування або розширений режим очікування) буде активовано інструкцією SLEEP.

3.7 Реалізація системи спостереження за станом здоров'я та алгоритму зниження енергоспоживання

Розробка програмного забезпечення для вбудованих систем повинна відповідати вимогам, що виділені нижче:

- код повинен бути надійним та робочим;
- можливість внесення змін та покращення системи;
- тесто-придатність системи;
- продуктивність та ефективність коду;
- повна працездатність на обраному пристрої.

Деяким пристроям на основі вбудованих системах необхідно працювати довгий час без можливості перезаряджати батареї, оновлення програмного забезпечення, або можливості перезагрузитися.

Програмне забезпечення розділене на дві частини. Перша – це частина управління Arduino, а друга необхідна для управління передачі даних по Wi-Fi ESP8266.

У ході написання програми були використано 7 бібліотек. Вони наведені у лістингу 3.1 та описані в пункті 3.6.

Лістинг 3.1 – Підключення необхідних бібліотек

```

#include <LiquidCrystal.h>
#include <Adafruit_MLX90614.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <TimeLib.h>

//energy
#include <avr/wdt.h>
#include <avr/sleep.h>

```

У лістингу 3.2 можна побачити реалізації основного циклу програми. Тут відбувається виклик усіх необхідних для роботи системи функцій, таких як `temperatureLM35()`, `ad8232()`, `temperatureMLX90614()`, `esp_8266()`. Вони відповідають за включення та роботу датчиків та модулю Wi-Fi. У цьому ж циклі відбувається частина реалізації алгоритму оптимізації. А саме виклик при необхідності функції для включення режиму сну.

Лістинг 3.2 – Реалізація основного циклу програми

```

void loop() {
  serialOutput();
  if (QS == true) // A Heartbeat Was Found
  {
    // BPM and IBI have been Determined
    // Quantified Self "QS" true when arduino finds a
    heartbeat
    fadeRate = 255; // Makes the LED Fade Effect Happen, Set
    'fadeRate' Variable to 255 to fade LED with pulse
    serialOutputWhenBeatHappens(); // A Beat Happened, Output
    that to serial.
    QS = false; // reset the Quantified Self flag for next
    time
  }
  temperatureLM35();
  delay(20); // take a break
  ad8232();
  delay(20); // take a break
  temperatureMLX90614();
  delay(20); // take a break
  esp_8266();
  delay(20); // take a break
  if (cmpEnergy())

```

```

    {
        if (!cmpData())
        {
            getData(SENSOR_1, 1);    getData(SENSOR_2, 2);
            getData(SENSOR_3, 3);
            sendData(SENSOR_1);    sendData(SENSOR_2);
            sendData(SENSOR_3);
            if (cmpData())
            {
                sleepModeESP();
                sleepModeArduino();
            }
        }
        else
        {
            sleepModeAll();
        }
    }
}

```

У наступному фрагменті кода показана реалізація функції порівняння необхідної для системи енергії з отриманою в даний момент часу від модуля отримання енергії.

Лістинг 3.3 – Реалізація функції порівняння необхідної енергії

```

bool cmpEnergy() {
    float amountOfSystemEnergy, systemEnergy;
    bool necessityToSleep;
    countUsingNowAmountOfSystemEnergy();
    necessityToSleep =
    cmpAmountSystemEnergyAndLTC3588Energy(systemEnergy);
    return necessityToSleep;
}

```

У лістингу 3.4-3.5 зображено переключення плати Seeeduno Mega та мікроконтролеру ESP до режиму сну. Для цього використовуються функції з бібліотеки `avr/sleep.h`. Відбувається вибір режиму сну у функції `set_sleep_mode(SLEEP_MODE_PWR_DOWN)` та переключення до цього режиму `sleep_mode()`. Для переключення ESP необхідно спочатку подати сигнал з Arduino, прийняти сигнал на модулі та перейти у режим сну за допомогою функції `ESP.deepSleep()`.

Лістинг 3.4 – Переключення плати Seeeduino Mega у режим сна

```

void sleepModeArduino() {
    wdt_enable(WDTO_2S); // Setup an interval to Watchdog timer
as 2 seconds
    WDTCSR |= (1 << WDIE); // Set WDIE bit for interruption
permit from WDT
    // Allow external interruption that will wake
up MC
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN),
emergencyCall, FALLING);
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // Setting up mode we
needed
    sleep_mode(); // Do sleep mode MC
    digitalWrite(LED_BUILTIN, stateLEDbuiltin);
    Serial.println("Arduino:- I am going for a Nap");
    timer(2);
    digitalWrite(LED_BUILTIN, LOW);

```

Лістинг 3.5 – Переключення мікроконтролеру ESP у режим сна

```

//Відправлення сигналу переходу у режим сну до ESP8266
void sleepModeESP() {
    digitalWrite(dataESPinfo, HIGH);
}
// перехід у режим сну ESP8266
void sleepModeESP() {
    Serial.println("deep sleep for 15 seconds");
    ESP.deepSleep(15e6);
}

```

У лістингу 3.6 показана реалізація функції `cmpData`. Дана функція отримує дані про те, як давно датчики послали останній раз отримані дані. Якщо, це відбувалося давно і система повинна скоро отримати нові дані, то перемикання у режим сну не відбувається, поки система не отримає нові дані від датчиків.

Лістинг 3.6 – Реалізація функції cmpData

```

bool cmpData() {
    time_t currentTime, differenceTime;
    getTime(currentTime);
    differenceTime = currentTime - SENSOR_1.time;
    if (differenceTime >= 10 && differenceTime < 20) // Request
for the data every 30 seconds
    {
        interruptableSleep(31000 - (differenceTime * 1000)); //
Difference sleep to the 30 seconds + 1 second to finish the
reading
    }
    struct sensor _sensor = {0, 0};
    getData(_sensor, 1);
    delay(1);

    float humanReadableTemp = (_sensor.iData / 1023.0) * 5.0 *
1000 / 10; // Get the human readable temperature
    if (humanReadableTemp > 37.7)
    {
        emergencyCall(); // Doctor intervention is needed
        return false;
    }
    differenceTime = currentTime - SENSOR_2.time;
    if (differenceTime >= 10 && differenceTime < 20) // Request
for the data every 30 seconds
    {
        interruptableSleep(31000 - (differenceTime * 1000)); //
Difference sleep to the 30 seconds + 1 second to finish the
reading
    }
    getData(_sensor, 2);
    delay(1);

    int humanReadableBreathTemp = (_sensor.iData / 1023) * 7 *
1000 / 10;
    if (humanReadableBreathTemp > 38)
    {
        emergencyCall(); // Doctor intervention is needed
        return false;
    }
    differenceTime = currentTime - SENSOR_3.time;
    if (differenceTime >= 10 && differenceTime < 20) // Request
for the data every 30 seconds
    {
        interruptableSleep(31000 - (differenceTime * 1000)); //
Difference sleep to the 30 seconds + 1 second to finish the
reading
    }
    getData(_sensor, 3);
}

```

```

delay(1);

if (_sensor.iData > 110)
{
    emergencyCall(); // Doctor intervention is needed
    return false;
}

return true;
}

```

Лістинг 3.7 – Реалізація функції отримання даних від датчиків

```

void getData(struct sensor& _sensor, uint8_t sensorNumber) {
    switch (sensorNumber)
    {
        case 1:
            _sensor.iData = analogRead(FIRST_SENSOR);
            break;
        case 2:
            _sensor.iData = analogRead(SECOND_SENSOR);
            break;
        case 3:
            _sensor.iData = analogRead(THIRD_SENSOR);
            break;

        default:
            break;
    }

    getTime(_sensor.time)
}

```

У лістингу 3.8 показана реалізація функцій системи спостереження за станом здоров'я. А саме ті функції, що безпосередньо приймають участь в отриманні даних з оточуючого середовища від датчиків. Тут показані функції `temperatureMLX90614()`, `ad8232()` та `temperatureLM35()`, що викликають датчики.

Лістинг 3.8 – Реалізація функцій роботи датчиків

```

void temperatureMLX90614() {

```

```

        Serial.print("Ambient                =                ");
Serial.print(mlx.readAmbientTempC());
        Serial.print("*C\tObject                =                ");
Serial.print(mlx.readObjectTempC()); Serial.println("*C");
        Serial.print("Ambient                =                ");
Serial.print(mlx.readAmbientTempF());
        Serial.print("*F\tObject                =                ");
Serial.print(mlx.readObjectTempF()); Serial.println("*F");
        Serial.println();
    }

    void ad8232() {
        // When have any signal -> "!" to the display

        if      ((digitalRead(ad8232L0minPin)      ==      1)      ||
(digitalRead(ad8232L0plusPin) == 1))
        {
            Serial.println('!');
        }
        else
        {
            // sending analog values
            Serial.println(analogRead(A0));
        }

        // delay to take values
        delay(10);
    }

    void temperatureLM35() {
        val = analogRead(tempPin);
        float mv = ( val/1024.0)*5000;
        float cel = mv/10;
        float farh = (cel*9)/5 + 32;
        Serial.print("TEMPERATURE = ");
        Serial.print(cel);
        Serial.print("*C");
        Serial.println();
        delay(1000);
    }
}

```

У лістингу 3.9 наведено підключення необхідних бібліотек для під'єднання до мережі передачі даних за допомогою модуля WiFi ESP8266.

Лістинг 3.9 – Підключення бібліотек ESP8266 WiFi

```

#include <ESP8266WiFi.h>
String apiWritekey = "*****";

```

```
char ssid[] = "XXXXXXXXXX"; // enter your wifi home router
ssid
char password[] = "XXXXXXXXXX" ; // enter your wifi home
router ssid
char server[] = "api.thingspeak.com";
```

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Аналіз розробленого програмного забезпечення

При розробці тестового програмного забезпечення було досягнуто наступних результатів:

- розроблене ПЗ дає змогу високоефективно використовувати накопичений заряд;
- є можливість замінювати одні елементи системи на інші, уникаючи при цьому великих збоїв системи;
- використано усі датчики та функції для повноцінної роботи системи.

Тестування програмного забезпечення відбувається на основі побудованої системи спостереження за станом здоров'я людини.

4.2 Результати тестування

Під час тестування розробленої системи було отримано декілька результатів при тестуванні в різних режимах роботи. Було проведено тестування без використання алгоритму з покращенням енергоефективності, а потім з використанням реалізованого алгоритму.

При проведенні первинного тестування було відключено функції керування сном системи. Спочатку вимірювалась енергоспоживання на мінімальній кількості елементів і поступово вони додавалися, щоб утворилась повноцінна система.

Для отримання результатів ефективності алгоритму оптимізації було проведено тестування за таких умов:

- час роботи системи вимірюється за один цикл отримування енергії з оточуючого середовища модулем LTC3588-1;
- один цикл заряджання дорівнює приблизно 60 секунд;

– чотири варіанти тестування апаратного забезпечення: Seeeduino Mega, Seeeduino Mega та датчики (MLX90614, AD8232, LM35x2), Seeeduino Mega та модуль Wi-Fi, Seeeduino Mega, модуль Wi-Fi та датчики (MLX90614, AD8232, LM35x2) , та отримання часу роботи для кожного варіанта при отримуванні енергії з оточуючого середовища модулем LTC3588-1;

– для проведення експериментального дослідження, в якості модулю Wi-Fi було використано ESP8266.

Результати тестування без використання алгоритму оптимізації показали, що система у першому варіанті тестування з використанням лише плати Seeeduino Mega, змогла працювати протягом більше 500 секунд. При підключенні датчиків, система працює трохи більше 180 секунд. Тестування роботи плати Seeeduino Mega з додатковою роботою модуля Wi-Fi показало роботу лише протягом 35 секунд. Це означає, що система не буде спроможна накопичувати енергію. При тестуванні повної системи відстеження здоров'я було виявлено, що система працює менше 30 секунд.

Таким чином, можна зробити висновок, що система спроможна працювати лише при використанні тільки мікроконтролера та датчиків. Використання модуля Wi-Fi стає неможливим, через занадто великі енергозатрати системи та неможливість накопичувати енергію. Без використання алгоритму оптимізації можна помітити, що система повноцінно працює з використанням тільки датчиків, при майже постійному отриманні енергії від модуля LTC3588-1. Для використання лише ESP8266 потрібне постійне отримання енергії, при цьому час від часу системі не вистачає струму та вона вимикається. Працездатність системи в повному обсязі майже неможлива.

Під час другого тестування, було додано алгоритм оптимізації енергоспоживання системи до коду програми та виміряно отримані результати заново з такими самими характеристиками.

Результати тестування з використанням алгоритму оптимізації енергоефективності системи показали, що досліджувана модель у першому варіанті тестування з використанням лише плати Seeeduno Mega, на основі мікроконтролеру ATmega 2560 змогла працювати протягом майже 950 секунд, що в 1,8 разів довше ніж при першому варіанті тестування. При підключенні датчиків, система працює близько 250 секунд, що в 1,4 рази довше. Тестування роботи плати Seeeduno Mega з модулем Wi-Fi показало роботу близько 80 секунд, що майже в 2,3 разів довше, та означає, що система зможе не тільки стабільно працювати, а й накопичувати енергію. При тестуванні повної системи відстеження здоров'я було виявлено, що система працює близько 70 секунд.

З використанням алгоритму оптимізації можна побачити суттєві зміни. При першому та другому варіанті система працює так само продуктивно, як і при першому тестуванні. Це не дивно, бо як було досліджено, найбільшу кількість енергії споживає основний мікроконтролер та модуль Wi-Fi. В даному варіанті тестування, системі вистачило отриманого струму для повноцінної роботи в повному складі необхідних елементів.

У процесі дослідження було порівняно працездатність системи з використанням алгоритму оптимізації робота/сон та без нього. Було отримано результати після тестування та доведена необхідність використання даного алгоритму при реалізації вбудованої системи з джерелом енергії на основі технології «energy harvesting».

ВИСНОВКИ

У ході виконання кваліфікаційної роботи проведено дослідження використання технології «energy harvesting» у вбудованій системі, що виконує роль моделі спостереження за здоров'ям людини. Виконано аналіз предметної області, виділено основні характеристики вбудованих систем та сформована узагальнена структура моделі з використанням «energy harvesting».

Було проаналізовано питання енергоспоживання в даного типу системах. Виділено обмеження на характеристики компонентів моделі та потужності джерел живлення, для практичного застосування у вбудованих системах. Приведено аналіз енергоспоживання компонентів та усієї моделі повністю.

Запропонований алгоритм зменшення енерговитратності системи за рахунок переведення в режим сну окремих елементів та/або системи повністю, при цьому не втрачаючи ефективності. Також, було переглянуто існуючі методи заощадження енергії, які залишають систему ефективною.

Було реалізовано систему спостереження за станом здоров'я людини з використанням алгоритму оптимізації енергоспоживання. При цьому було проведено дослідження ефективності даного алгоритму та приведено результати тестування.

Дослідження показало перспективність використання технології «energy harvesting» у вбудованих системах, особливо при використанні малопотужних мікропроцесорів.

Подальші дослідження в області включають в себе реалізації інших оптимізованих алгоритмів, для можливості використання «energy harvesting» в більш потужних вбудованих системах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Energy Harvesting Technologies/Shashank Priya, Daniel J. Inman - 2009 - 517с.
2. Internet of things (iot): concepts and applications/ mansaf alam; kashish ara shakil; samiya khan - cham, switzerland : springer, [2020]
3. Energy harvesting: енергія з нічого [електронний ресурс] // «Компьютера». 132 інфраструктурні галузі економіки
4. Бочарніков і. Energy harvesting. Новий етап у розвитку автономних пристроїв / І. Бочарніков, М. Уїтакер // Компоненти та технології. - 2010. - № 8. - с. 146-149.
5. Energy harvesting – others [електронний ресурс]. - режим доступу: http://www.ti.com/solution/energy_harvesting – texas instruments (дата звернення: 15.07.22).
6. Romero-ramirez, edwar. «energy harvesting from body motion using rotational micro-generation», dissertation, michigan technological university, 2010 [електронний ресурс]. – режим доступу: <http://digitalcommons.mtu.edu/ etds/404>.
7. Искусство схемотехники/ХоровицП., Хилл У. - Пер с англ – Изд 2 е. – М.: Издательство БИНОМ – 2014 – 704с.
8. A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, H. Owen, Wireless Sensor Networks for Rehabilitation Application: Challenges and Opportunities, J. of Network and Computer Applications 36 (2013) 1–15.
9. <https://www.computerweekly.com/de/definition/Embedded-System-Eingebettetes-System>
10. Pandharipande, A.; Caicedo, D.;Wang, X. Sensor-driven wireless lighting control: System solutions and services for intelligent buildings. IEEE Sens. J. 2014, 14, 4207–4215. [CrossRef]

11. Vullers, R. Energy Harvesting for Autonomous Wireless Sensor Networks [Текст] / R. Vullers, R. Schaijk, H. Visser, J. Penders, C. Hoof // IEEE Solid-State Circuits Magazine. – 2010. – 2(2). – С. 29–38.
12. Sudevalayam, S. Energy Harvesting Sensor Nodes: Survey and Implications [Текст] / S. Sudevalayam, P. Kulkarni // IEEE Communications Surveys & Tutorials. – 2011. – Т. 13, №3. – С. 443–461.
13. Harb, A. Energy harvesting: State-of-the-art [Текст] / A. Harb // Renewable Energy. – 2011. – Т. 36, №10. – С. 2641–2654.
14. Vivek Tiwari, Sharad Malik, and Andrew Wolfe, “Power Analysis of Embedded Software: A First Step Towards Software Power Minimization,” IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Vol.2, No.4, pp. 437-445, December 1994
15. N. Kavvadias, P. Neofotistos, S. Nikolaidis, K. Kosmatopoulos and Th. Laopoulos, “Measurement Analysis of the Software-Related Power Consumption in Microprocessors”, IEEE Trans. on Instrumentation and Measurement, Vol.53.N.4, August 2004
16. Ruan, T. Energy-Aware Approaches for Energy Harvesting Powered Wireless Sensor Nodes [Текст] / T. Ruan, Z. J. Chew, M. Zhu. // IEEE Sensors Journal. – 2017. – Т. 17, №7. – С. 2165–2173.
17. [IEEE Transactions on Instrumentation and Measurement](#) 57(4)(4):797 – 804 DOI: [10.1109/TIM.2007.913724](https://doi.org/10.1109/TIM.2007.913724)
18. Internet of things: wireless sensor networks [Текст] / IEC – Geneve : IEC, 2014. – 78 с. – (White paper). – ISBN 978-2-8322-1834-1.
19. Harb, A. Energy harvesting: State-of-the-art [Текст] / A. Harb // Renewable Energy. – 2011. – Т. 36, №10. – С. 2641–2654.
20. Rault T. Energy efficiency in wireless sensor networks: A top-down survey [Текст] / T. Rault, A. Bouabdallah, Y. Challal // Computer Networks. – 2014. – Т. 67. – С. 102–131.

21. T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. Lim, A. Terzis, A. Watt, J. Jeng, B.-r. Chen, K. Lorincz, M. Welsh, Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results, in: IEEE Conference on Technologies for Homeland Security, Waltham, 2008, pp. 187–192.
22. R. Fensli, E. Gunnarson, T. Gundersen, A Wearable ECGRecording System for Continuous Arrhythmia Monitoring in a Wireless Tele-Home-Care Situation, in: 18th IEEE Symp. on Computer-Based Medical Systems, 2005, pp. 407–412.
23. J. Chen, K. Kwong, D. Chang, J. Luk, R. Bajcsy, Wearable sensors for reliable fall detection, in: 27th IEEE Conference of the Engineering in Medicine and Biology Society, Shanghai, 2005, pp. 3551–3554.
24. A. Hadjidj, A. Bouabdallah, Y. Challal, Rehabilitation supervision using wireless sensor networks, in: IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Lucca, Italy, 2011, pp. 1–3.
25. W.-W. Chang, T.-J. Sung, H.-W. Huang, W.-C. Hsu, C.-W. Kuo, J.-J. Chang, Y.-T. Hou, Y.-C. Lan, W.-C. Kuo, Y.-Y. Lin, Y.-J. Yang, A smart medication system using wireless sensor network technologies, *Sensors and Actuators A: Physical* 172 (2011) 315–321.
26. H. Ren, M. Meng, X. Chen, Wireless Assistive Sensor Networks for the Deaf, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, 2006, pp. 4804–4808.
27. H. Alemdar, C. Ersoy, Wireless sensor networks for healthcare: A survey, *Computer Networks* 54 (2010) 2688–2710.
28. J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, M. Welsh, Wireless Sensor Networks for Healthcare, *Proceedings of the IEEE* 98 (2010) 1947–1960.
29. Сало А. М., Кравець О. І., «Методи та засоби оптимізації енергоспоживання вендингових автоматів.» – р.2015