

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»
СИСТЕМА ДИСТАНЦІЙНОГО ЗЧИТУВАННЯ ПОКАЗНИКІВ
ЛІЧИЛЬНИКІВ НА ОСНОВІ ІОТ
IOT-BASED REMOTE METER READING SYSTEM**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Церпіш Ігор Сергійович

(підпис)

Керівник:
к.т.н., доцент
Костючко Сергій Миколайович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 04 » червня 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. ТЕРЛЕЦЬКИЙ

« 10 » 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Церпін Ігор Сергійович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Система дистанційного зчитування показників лічильників на основі IoT*

Керівник роботи *к.т.н., доцент Костючко Сергій Миколайович*

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *11.06.2025р.*

3. Вихідні дані до роботи *джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування.*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Огляд літератури та аналітичної частини предметної області

Обґрунтування вибору засобів та методів реалізації

Практична реалізація системи дистанційного зчитування показників лічильників на основі IoT

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Скріншоти існуючих засобів обліку

Скріншоти налаштувань ESP32

Архітектура системи дистанційного зчитування

Схематичне зображення роботи системи дистанційного зчитування

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналітична частина предметної області</i>	<i>Костючко С.М., доцент</i>		
<i>Обґрунтування вибору засобів та методів реалізації</i>	<i>Костючко С.М., доцент</i>		
<i>Практична реалізація пристрою</i>	<i>Костючко С.М., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2025 р.	Виконано
2.	<i>Аналітична частина предметної області, обґрунтування вибору засобів та методів реалізації</i>	до 02.03.2025 р.	Виконано
3.	<i>Практична реалізація пристрою</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 15.05.2025 р.	Виконано
9.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
10.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
11.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедрі</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

_____ (підпис)

Церпіш І.С.

_____ (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Костючко С.М.

_____ (прізвище, ініціали)

АНОТАЦІЯ

Церпіш І.С. Система дистанційного зчитування показників лічильників на основі IoT.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Перший розділ присвячено огляду предметної області, розглядаються принципи роботи сучасних засобів дистанційного зчитування показів лічильника, описано актуальні вирішення проблем обліку водоспоживання. Здійснено огляд існуючих аналогів, також сформульовано основні вимоги до майбутньої системи та обґрунтовано напрямок розробки.

В другому розділі здійснено вибір технологій та засобів реалізації. Розглянуто можливі варіанти апаратного забезпечення, мережевих протоколів та програмних рішень.

Третій розділ складається з розробки пристрою для забезпечення дистанційного зчитування показів з лічильників. Розроблено серверну частину для фіксування результатів дистанційного зчитування.

Ключові слова: лічильники, дистанційне зчитування, ESP32, arduino, мікроконтролер, плата, OV5640, інтернет речей.

ANNOTATION

Tserpish I. Iot-based remote meter reading system. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions, a references, three appendices.

The first section is devoted to an overview of the subject area, the principles of operation of modern means of remote meter reading are considered, current solutions to water consumption accounting problems are described. An overview of existing analogues is carried out, the main requirements for the future system are also formulated and the direction of development is justified.

In the second section, a selection of technologies and means of implementation is made. Possible options for hardware, network protocols, and software solutions are considered.

The third section consists of the development of a device for remote meter reading. A server part has been developed for recording remote reading results.

Keywords: meters, remote reading, ESP32, arduino, microcontroller, board, OV5640, internet of things.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІТИЧНА ЧАСТИНА ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Огляд та дослідження літератури.....	9
1.2 Особливості та різновид засобів обліку	12
1.3 Огляд сучасних систем дистанційного зчитування показників лічильників	15
1.4 Аналіз технологій Інтернету речей (IoT) у сферах дистанційного моніторингу	18
РОЗДІЛ 2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ТА МЕТОДІВ РЕАЛІЗАЦІЇ.....	21
2.1 Особливості використання апаратного забезпечення	21
2.2 Аналіз протоколів передачі даних LoRaWAN, NB-IoT, MQTT.....	27
2.3 Особливості використання програмних технологій.....	33
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРИСТРОЮ.....	36
3.1 Налаштування апаратного та програмного забезпечення	36
3.2 Схема підключення та алгоритм роботи пристрою	40
3.3 Тестування та оптимізація пристрою.....	42
ВИСНОВКИ.....	47
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТКИ.....	52

ВСТУП

Працюючи у сфері водопостачання можна виділити низку проблем з якими стикаються люди. Однією з них є передача показів лічильника, здається банальна справа, потребує лише декілька хвилин часу, процедура здійснюється здебільшого раз на місяць але іноді приносить дискомфорт. Забули передати, важкодоступне місце, лінь, та багато іншого може завадити для споживача.

Спробувати вирішити цю проблему можна за допомогою системи дистанційного зчитування показників лічильників. Даний пристрій за рахунок своєї простоти і невеликої собівартості зможе дозволити звичайний споживач, для моніторингу своїх витрат. З використанням нашого приладу обліку, можна суттєво спростити збір інформації.

Актуальність теми полягає в тому, що з розвитком технологій IoT, зростає потреба в автоматизації процесів збору даних, що дозволяє підвищити ефективність та точність роботи різноманітних систем. Однією з таких систем є дистанційне зчитування показників лічильників, вона дає змогу зменшити витрати часу на обробку даних, знизити людський фактор та забезпечити більш точну інформацію для контролю витрат ресурсів. Також розробка цього приладу може суттєво змінити економічні затрати споживача, забезпечуючи йому стабільну передачу цифр, що в свою чергу зменшує шанси стати боржником перед підприємствами.

Метою роботи є розробка системи дистанційного зчитування показників лічильників на основі технології інтернету речей (IoT), що дозволить спростити процес збору та передачі даних з лічильників до централізованої бази даних.

Об'єктом дослідження є системи дистанційного зчитування показників лічильника.

Предмет дослідження – технології та методи реалізації системи дистанційного зчитування показників лічильників на основі IoT.

Завдання, які необхідно виконати:

- запропонувати актуальне вирішення сучасних проблем з передачею показників;
- реалізувати систему збору та передачі показників лічильників;
- розробити пристрій для зчитування показників та серверну частину для зберігання даних;
- дослідити ефективність обраних компонентів пристрою та мережевої частини.

Результати роботи доповідалися на міжнародній науково-практичній конференції молодих вчених та студентів «Програмне та апаратне забезпечення в інформаційних технологіях», Луцьк, 6 травня 2025 року [1].

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд та дослідження літератури

Термін «Інтернет речей» вперше був згаданий Кевіном Ештоном у 1999 році, коли він працював над оптимізацією постачання товарів у Procter & Gamble. Він пояснив ідею, що пристрої можуть самостійно збирати та передавати інформацію без необхідності втручання людини, що стало основою для подальшого розвитку цієї технології. Ідея була занадто цікавою, щоб залишити її в такому вузькому контексті, тому за останнє десятиліття вона розповсюдилася, охопивши широкий спектр застосувань, таких як охорона здоров'я, комунальні послуги, фінанси, транспорт. З того часу слово «речі» змінило своє значення, але основна ідея IoT залишилась тією ж, вона здатна організувати середовище для взаємодії, в якому обчислювальні технології зможуть комунікувати одна з одною, налаштовуватись під потреби користувачів та в той же час працювати автономно, збираючи інформацію з зовнішніх джерел.

У 2000-х роках IoT почав набувати розголосу, з появою компактних мікроконтролерів, таких як серії AVR і ARM, стало можливим створення невеликих автономних пристроїв. У цей час з'являються перші побутові пристрої з Wi-Fi-зв'язком але вони мали високе енергоспоживання, що обмежувало їх автономність, але все ж це був значний крок у розвитку технологій, та важливий прогрес у програмуванні мікроконтролерів [2].

Справжній прорив стався у 2010-х роках, коли ринок наситився недорогими модулями зв'язку, такими як Wi-Fi, Bluetooth, GSM і особливо Bluetooth Low Energy, який був розроблений спеціально для пристроїв з обмеженим енергоспоживанням. Саме цей протокол став основою у сфері IoT-пристроїв, систем моніторингу здоров'я, трекерів та «розумного» дому.

Паралельно розвивалися і технології з великою зоною покриття, зокрема, LPWAN (Low Power Wide Area Networks), включаючи LoRa, Sigfox і NB-IoT. Ці протоколи дозволяли об'єднувати пристрої в мережі, які можуть працювати

місяцями або навіть роками від батареї, забезпечуючи при цьому передачу даних на десятки кілометрів.

На межі 2020-х років почалося масове впровадження IoT у промисловість, транспорт, міську інфраструктуру та енергетику. Це стало можливим завдяки удосконаленню хмарних сервісів, аналітики даних та розвитку 5G, який доповнив наявні рішення наднизькою затримкою та високою пропускнуою здатністю.

Сьогодні екосистема IoT охоплює мільярди пристроїв беручи початок від простих сенсорів до складних вбудованих систем. Їх поєднує здатність збирати, передавати та обробляти дані з метою оптимізації процесів, підвищення безпеки, енергоефективності та зручності в повсякденному житті. Регулярно проводяться дослідження для покращення ефективності використання IoT.

Стаття «An overview of IoT architectures, technologies, and existing open-source projects», автори якої є Томас Домінгес-Боланьо, Омар Кампос, Валентін Баррал, Карлос Ескудеро та Хосе А. Гарсія-Нає, проведено всебічний огляд архітектур, технологій та відкритих проєктів у сфері Інтернету речей [3]. Це дослідження охоплює важливі аспекти розвитку IoT-систем, включаючи різноманітні архітектурні моделі, основні протоколи передачі даних та доступні програмні рішення, що використовуються в сучасних IoT-екосистемах.

Автори детально аналізують різні архітектурні моделі IoT, зокрема багаторівневі структури, що включають пристрої, мережі передачі даних, хмарні платформи та аналітичні сервіси. Вони підкреслюють необхідність інтеграції цих компонентів для забезпечення ефективної та безпечної роботи систем IoT. Такий підхід дозволяє забезпечити максимальну ефективність у зборі даних, їх передачі та обробці, що є критично важливим для розвитку смарт-мереж, таких як енергетичні або водопостачальні системи [3].

Дослідження також охоплює технології та протоколи, що лежать в основі IoT-комунікацій, зокрема MQTT, CoAP, LoRaWAN та NB-IoT. Автори здійснюють порівняльний аналіз цих протоколів, оцінюючи їх переваги та недоліки. Наприклад, LoRaWAN та NB-IoT виявляються надзвичайно

ефективними для створення масштабованих і енергоефективних IoT-рішень для віддаленого моніторингу, таких як системи дистанційного зчитування показів лічильників.

Ще однією важливою темою є огляд відкритих платформ та проєктів IoT, таких як ThingSpeak, OpenHAB та Home Assistant. Автори зазначають, що ці відкриті рішення дозволяють розробникам ефективно створювати IoT-системи з низькими витратами, забезпечуючи при цьому високу гнучкість та масштабованість. Вони оцінюють функціональність і сумісність цих платформ, що має велике значення для подальшого розвитку та впровадження IoT у різних сферах.

Автори статті акцентують увагу на кількох важливих аспектах, які потребують подальшого вдосконалення для розвитку IoT. Зокрема, автори відзначають необхідність стандартизації протоколів, підвищення рівня безпеки даних і інтеграції IoT із такими передовими технологіями, як штучний інтелект і блокчейн. Вони рекомендують застосовувати відкриті стандарти та платформи для забезпечення інтероперабельності різних IoT-систем, що допоможе прискорити інновації та розширити можливості IoT-рішень у різних галузях.

У 2025 році міжнародна команда дослідників, включаючи Канаа Алаа Аль-Саммака та інших, запропонувала алгоритм для оптимізації енергоспоживання в смарт-лічильниках на основі IoT, використовуючи технології LoRaWAN та NB-IoT. Їхній підхід дозволив зменшити споживання енергії та кількість переданих пакетів даних, що сприяє підвищенню ефективності систем дистанційного зчитування [4].

У 2024 році дослідники з Індії, зокрема Шрікант Патіл та інші, розробили систему енергомоніторингу на основі IoT, яка дозволяє віддалено контролювати споживання енергії в реальному часі. Їхнє дослідження показало високу точність та мінімальну похибку в показах, що сприяє підвищенню обізнаності користувачів про власне споживання енергії [5].

У 2022 році дослідники з Німеччини, зокрема Кнайер та Кривінська, провели систематичний огляд технологій смарт-лічильників для ефективного

управління енергією в домогосподарствах та організаціях. Вони проаналізували 47 наукових статей, виявивши, що впровадження цих технологій у Німеччині ще не охоплює велику кількість користувачів, хоча потреба в підвищенні енергоефективності постійно зростає. Дослідники відзначили, що на прийняття смарт-лічильників впливають як технологічні, так і організаційні та екологічні фактори, і наголосили на необхідності подальших досліджень для масштабування в різних секторах [6].

У 2023 році було проведено дослідження у середземноморських країнах, яке зосереджувалося на впливі смарт-лічильників на енергоефективність домогосподарств з низьким рівнем доходу. Автори дослідження проаналізували споживання енергії до та після встановлення смарт-лічильників і виявили суттєве зниження енергоспоживання завдяки кращому моніторингу та управлінню. Особливо відзначено, що такі технології допомагають значно економити ресурси у вразливих верствах населення, що робить доцільним розширення програм впровадження смарт-лічильників для досягнення ширшої енергоефективності [7].

1.2 Особливості та різновид засобів обліку

Засоби обліку споживання ресурсів, таких як електроенергія, газ, вода, тепло є ключовими елементами сучасного побуту та промисловості. Вони дозволяють здійснювати ефективний та точний контроль використаних ресурсів. Завдяки таким системам обліку споживачі мають змогу проводити аналіз споживання та контролювати свої витрати. А для комунальних підприємств лічильники є чудовим інструментом для спостереження за обліком ресурсів та запобігання несанкціонованому їх споживанню.

Нижче на рисунку 1.1 наведено фото лічильника холодної води. Завдяки впровадженню таких технологій можна більш раціонально використовувати природні запаси та посприяти їх вичерпанню у майбутньому.



Рисунок 1.1 – Лічильник холодної води

Виділяють декілька ключових видів лічильників:

1) лічильники електроенергії:

– механічні лічильники – працюють на основі електромагнітної індукції, мають тривалий термін служби та стійкі до перепадів напруги. Однак, їх точність може бути нижчою порівняно з електронними аналогами;

– електронні лічильники – використовують цифрові технології для вимірювання витрат електроенергії, мають високу точність та підтримують багатотарифний облік (рис. 1.2);

– смарт-лічильники – оснащені модулями зв'язку, що дозволяють автоматично передавати дані до централізованої бази, підтримують зонний облік, що дозволяє значно економити при споживанні електроенергії вночі;



Рисунок 1.2 – Електронний лічильник електроенергії [8]

2) лічильники газу:

- механічні лічильники – найбільш поширені, працюють на основі вимірювання об'єму газу, що проходить через пристрій (рис. 1.3);
- смарт-лічильники газу – новітні пристрої, які автоматично ведуть облік спожитого газу та передають показання оператору газорозподільної мережі, забезпечуючи зручність для споживачів;



Рисунок 1.3 – Smart-лічильник газу [8]

3) лічильники води:

- тахометричні лічильники – вимірюють об'єм води за допомогою обертання крильчатки або турбіни під дією потоку води;
- ультразвукові лічильники – визначають витрату води за допомогою ультразвукових сигналів, відрізняються високою точністю та своєю довговічністю (рис. 1.4).



Рисунок 1.4 – Ультразвуковий лічильник води [8]

1.3 Огляд сучасних систем дистанційного зчитування показників лічильників

Одна з найпоширеніших проблем водоканалів і прикладних компаній – облік ресурсів. Багато постачальників досі ведуть облік води вручну, показники лічильників повідомляють самі абоненти або фіксація відбувається за допомогою служб контролю. Такий метод незручний і все швидше втрачає свою актуальність.

Завдяки впровадженню сучасних технологій дистанційного зчитування даних, зменшується потреба в ручному зборі показників, що підвищує точність обліку та знижує операційні витрати. Впровадження розумних лічильників дозволяє автоматизувати процеси, роблячи їх зручнішими для користувачів та ефективнішими для керуючих компаній.

Уже сьогодні на ринку існують компанії, яким вдалося спроектувати різноманітні технології для дистанційного зчитування та передачі даних, що стали невід'ємною частиною для моніторингу ресурсів. Всі вони приклали чимало зусиль розробляючи рішення, що дозволяють не лише автоматизувати процеси збору даних, а також здійснювати аналіз споживання, задля контролю

витрат. Компанії пропонують системи вирішення проблем для великих підприємств, так і компактні для звичайного споживача.

Нижче розглянемо одні з найпопулярніших систем дистанційного зчитування показників вітчизняного виробництва та лідерів зарубіжного ринку.

«Jooby» – український виробник, лідер нашого ринку серед пристроїв для моніторингу водоспоживання на основі технології LoRaWAN, що забезпечують енергоефективний і стабільний зв'язок. Принцип роботи здійснюється за рахунок радіомодулів, які кріпляться на прилади обліку та фіксують показники із заданою частотою та оптичних датчиків, що зчитують показники з циферблата водоміра (рис. 1.5).



Рисунок 1.5 – Оптичний радіомодуль для лічильників води Novator [9]

Показники зберігаються в тимчасовій пам'яті пристрою – щоденних, щотижневих, місячних і річних журналах. Паралельно радіомодулі надсилають дані через радіохвилі на базову станцію, використовуючи технологію LoRaWAN®. Звіти показники надходять каналами інтернет-зв'язку на головний сервер [9].

Компанія «Антап Україна» також пропонує системи зчитування даних із лічильників, зокрема, системи на базі Радіо та M-Bus. Ці системи дозволяють

дистанційно знімати показники з лічильників, встановлених у важкодоступних місцях.

M-Bus система призначена для організації збору даних з різних вимірювальних приладів в місцях високої концентрації, таких як, багатоквартирні житлові будинки, промислові об'єкти. Система дозволяє дистанційно змінювати налаштування регулятора керування, що забезпечує повне віддалене керування системою опалення та водопостачання. Розміщення системи в хмарному сервері забезпечує високу швидкість роботи, необмежену кількість користувачів, високий рівень захисту інформації [10]. На рисунку 1.6 зображено схему роботи системи M-Bus.

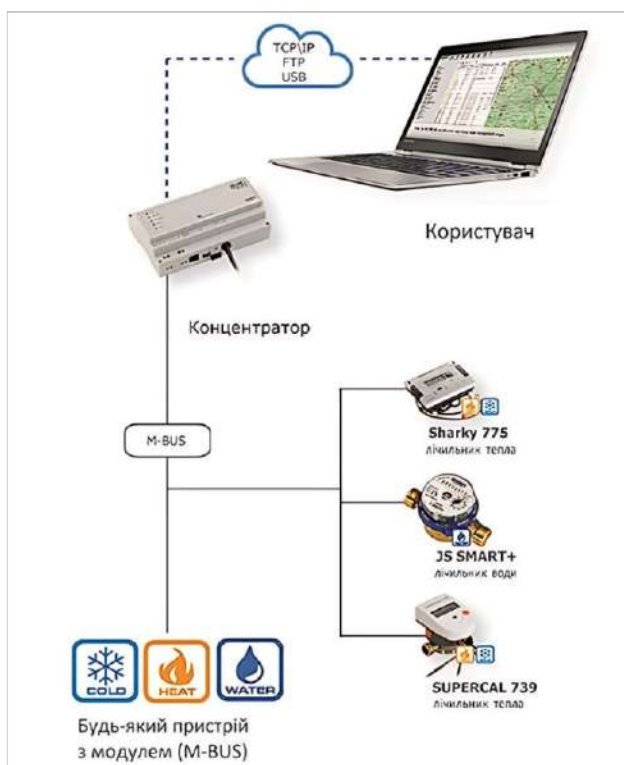


Рисунок 1.6 – Дротова передача даних системою M-Bus [10]

Itron – американська компанія, яка спеціалізується на розробці рішень для обліку та управління енергоресурсами. Вона пропонує інтелектуальні лічильники та системи для дистанційного зчитування даних, що дозволяють комунальним підприємствам ефективно керувати споживанням електроенергії, води та газу. Компанія пропонує комплексні методи вирішення питань з

використанням електроенергії, газу, води та теплової енергії, включаючи вимірювальні пристрої, комунікаційні системи та програмне забезпечення [11].

Однією з ключових технологій Itron є автоматизоване зчитування показників AMR, яке дозволяє комунальним підприємствам ефективно збирати дані без необхідності фізичного доступу до них. Платформа Temetra збирає, оптимізує та надсилає дані лічильників працівникам, а потім повертає зібрані дані керівникам для моніторингу.

На рисунку 1.7 показано цифровий лічильник нового покоління для дистанційного зчитування даних від компанії Itron.



Рисунок 1.7 – Цифровий лічильник дистанційного зчитування даних Itron [11]

1.4 Аналіз технологій Інтернету речей (IoT) у сферах дистанційного моніторингу

Поняття технології Інтернет речей несе за собою сукупність різноманітних систем, що працюють між собою в одній мережі та взаємодіють з навколишнім світом за допомогою спеціальних каналів зв'язку, які в кінцевому результаті з'єднуються в єдиний протокол доступу до глобальної мережі. Під глобальною

мережею для Інтернету речей мається на увазі мережа Інтернет, а протоколом доступу є IP.

З розвитком Інтернету речей все більше предметів буде оснащено цифровою підтримкою, що може значно вдосконалити систему безпеки, медицини, освіти, аналітики і управління, та покращити перспективи у забезпеченні кращої якості життя населення. Простими словами, технології Інтернет речей – це взаємопов’язана система об’єктів, пристроїв, машин та приладів, що оснащені датчиками, які дають можливості обмінюватися даними у мережі Інтернет (рис. 1.8). Це одна з технологій, якій притаманний дуже швидкий розвиток та надає величезні можливості для суспільства та уряду [12].



Рисунок 1.8 – Ілюстрація використання Інтернету речей

Можна виділити декілька ключових видів Інтернет речей:

– споживчий, це певний вид пристроїв, що використовується людьми для підвищення комфорту у повсякденному житті, забезпечує автоматизацію цифрових технологій. Сюди можна віднести Wi-Fi та Bluetooth технології такі як розумні годинники, фітнес-браслети, розумні окуляри та багато іншого;

– промисловий, є найдинамічнішим та найважливішим крилом у сфері Інтернет речей, зосереджується в основному на розширенні існуючих

промислових систем, роблячи їх більш продуктивними та ефективними. Використовується зазвичай на великих фабриках, заводах, часто пов'язане з такими галузями, як охорона здоров'я, сільське господарство, автомобільна промисловість і логістика;

– інфраструктурний, передбачає розвиток технологій IoT, які забезпечують покращення надійності, безпеки та ефективності систем, які надають базові потреби для населення, а саме транспорт, енергію, воду, зв'язок. Цей тип IoT тісно пов'язаний із промисловим інтернетом речей але акцент робиться саме на масштабних об'єктах, що мають критичне значення для функціонування міста або регіону;

– військовий, зосереджений в основному на використанні IoT технологій для покращення функціонування у військових умовах. Існують програми, що включають з'єднання кораблів, літаків, танків, солдатів, безпілотників і навіть передових оперативних баз через взаємопов'язану систему;

– медичний, це сукупність пристроїв та систем, які допомагають здійснювати діагностику, моніторинг стану здоров'я пацієнта та контролювати роботу медичних закладів. Лікарі можуть дистанційно стежити за станом здоров'я пацієнта, реагувати на критичні зміни в режимі реального часу та вчасно приймати важливі рішення. Це особливо актуально для пацієнтів з хронічними захворюваннями, які потребують постійного нагляду.

Нашу розробку можна віднести до інфраструктурного типу, оскільки, лічильники є частиною комунальної інфраструктури. Автоматизація зчитування показників надає змогу комунальним підприємствам ефективніше керувати ресурсами та контролювати витрати.

Впровадження технології Інтернету речей для дистанційного обліку ресурсів радикально змінило підхід муніципалітетів до керування своїми комунальними послугами. Дані в реальному часі, зниження витрат на робочу силу, підвищення точності виставлення рахунків, попереджувальна система обслуговування та аналіз даних, є основними перевагами, які пропонує Інтернет речей.

РОЗДІЛ 2

ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ТА МЕТОДІВ РЕАЛІЗАЦІЇ

2.1 Особливості використання апаратного забезпечення

Основними функціональними компонентами у розробці системи дистанційного зчитування показників лічильника будуть виступати:

- мікроконтролер ESP32-CAM;
- камера OV5640 66 degrees;
- блок живлення 5V;
- світлодіод.

Однією з ключових складових системи дистанційного зчитування показників лічильників є апаратна частина, яка забезпечує збір даних. Для зчитування даних, було обрано мікроконтролер ESP32 у комплекті з камерою OV5640 66 KIT (рис. 2.1). Такий вибір зумовлений рядом технічних переваг, а також відповідає вимогам до енергоефективності, вартості, функціональності та простоти інтеграції. Завдяки сумісності з різними периферійними модулями, ESP32 з камерою легко адаптується під конкретні потреби системи.



Рисунок 2.1 – Мікроконтролер ESP32 у комплекті з камерою OV5640 66 KIT

Нижче представлена таблиця 2.1 з характеристиками мікроконтролера ESP32-CAM.

Таблиця 2.1 – Характеристики мікроконтролера ESP32-CAM [13]

Параметр	Значення
Корпус	SMD-монтаж
Модель	OV5640 ESP32 CAM
Споживана потужність	Глибокий сон: 6 мА при 5 В; Модемний сон: 20 мА при 5 В; Легкий сон: 6,7 мА при 5 В
Особливості	Підтримка автофокусу OV5640, Вбудований світлодіодний спалах; підтримка завантаження зображень через Wi-Fi;
Напруга живлення	5 В, 2 А
Інтерфейс	USB Type-C
Температурний режим	Від -20°C до 85°C
Слот для карти пам'яті	Підтримка TF-карт до 32 ГБ
Пам'ять	Вбудована 520 КБ SRAM, зовнішня 4 МБ PSRAM
Процесор	32-бітний низькоспоживчий CPU з тактовою частотою до 160 МГц, продуктивністю до 600 DMIPS

2.1.1 Мікроконтролер ESP32-CAM

Плата ESP32-CAM є багатофункціональним мікроконтролером, який об'єднує в собі потужний процесор ESP32, камеру та можливості бездротового зв'язку. Вона стала надзвичайно популярною серед розробників у сфері Інтернету речей завдяки своїм компактним розмірам (рис. 2.2), низькому енергоспоживанню та широкому спектру функцій.

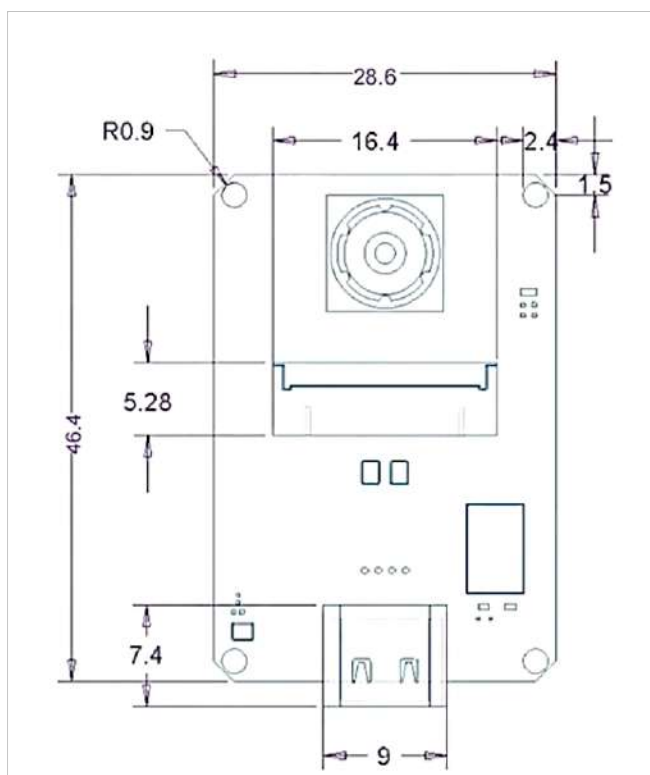


Рисунок 2.2 – Розмір мікроконтролера ESP32-CAM

Модуль оснащений процесором ESP32-D0WDQ6 з двома ядрами Tensilica LX6, що працює на частоті до 240 МГц та підтримує до 8 МБ PSRAM, що дозволяє обробляти зображення та відео у реальному часі.

У цій моделі використовується оновлена камера OV5640, яка забезпечує підвищену стабільність, кращу якість зображення та менше тепловиділення під час роботи. Камера підтримує апаратний автофокус, що особливо важливо для фіксації чітких зображень з різної дистанції, а також володіє широким кутом огляду. Модуль може працювати без додаткового контролера бо в ньому реалізована повна функціональність системи з живленням, обробкою та передачею даних.

У тестовому режимі фото з камерою OV5640 можна отримувати зі швидкістю до 30 кадрів на секунду у форматі VGA, а вища роздільна здатність дозволяє отримувати більш деталізовані знімки з високою чіткістю.

Вбудована Flash-пам'ять містить 32 Мбіт, а також підтримується підключення TF-карти об'ємом до 32 ГБ, що значно розширює можливості щодо зберігання відео- або фотоматеріалів.

Плата підтримує як живлення через USB, так і зовнішнє живлення через літієву батарею. У схемі передбачено енергозберігаючий контролер живлення IP5306, який відповідає за керування живленням, заряджанням батареї та її вмиканням та вимиканням.

Плата підтримує автоматичне завантаження прошивки через USB-інтерфейс завдяки чіпу CH552. Це дає змогу значно спростити прошивку та налагодження пристрою без необхідності використання додаткового UART-конвертера. Завдяки використанню ядра E8051 з підвищеною швидкістю обробки команд, завантаження мікропрограми здійснюється у 8-15 разів швидше, ніж у класичних рішеннях на основі контролера MCS51. Також підтримується повний набір стандартних інтерфейсів для розширення: I2C, SPI, UART, PWM, ADC, що дозволяє інтегрувати додаткові датчики, керуючі елементи, модулі зв'язку [14].

PSRAM у платі дозволяє використовувати фреймворк розпізнавання обличчя ESP-WHO від Espressif. Це відкриває додаткові можливості застосування у системах контролю доступу, обліку персоналу, автоматизації торгівлі, моніторингу та безпеки. Оскільки, камера має апаратну підтримку фокусування та ISP функції, зображення автоматично коригуються за параметрами світла, контрасту, шуму та експозиції, що суттєво підвищує стабільність роботи системи у складних умовах освітлення.

ESP32-CAM підтримує стандарт Wi-Fi 802.11 b/g/n, а також Bluetooth, що забезпечує бездротову передачу даних на сервер, мобільний додаток або інші пристрої. Також модуль має можливість зберігати фото- та відеоматеріали на карті microSD, що розширює можливості автономної роботи системи [14].

Ще однією особливістю ESP32-CAM є підтримка режиму глибокого сну, де споживання струму зменшується до 6 мА, що робить модуль придатним для енергоефективних пристроїв з автономним живленням.

Програмне забезпечення для роботи з модулем може розроблятися з використанням Arduino IDE, Esp-IDF, VScode або MicroPython, що забезпечує

велику гнучкість і доступність для розробників різного рівня підготовки. На рисунку 2.3 зображено компоненти описаного мікроконтролера.

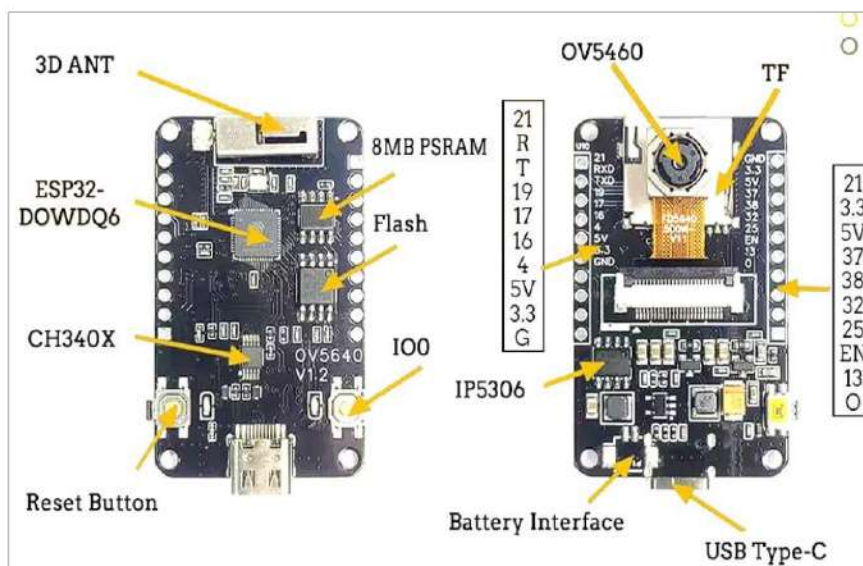


Рисунок 2.3 – Компоненти мікроконтролера ESP32-CAM

Також, нижче можна побачити схему підключення мікроконтролера ESP32, структурну побудову пристрою, що дозволяє краще зрозуміти логіку з'єднань усіх апаратних компонентів, камери, SD-карти, також інших датчиків (рис. 2.4).

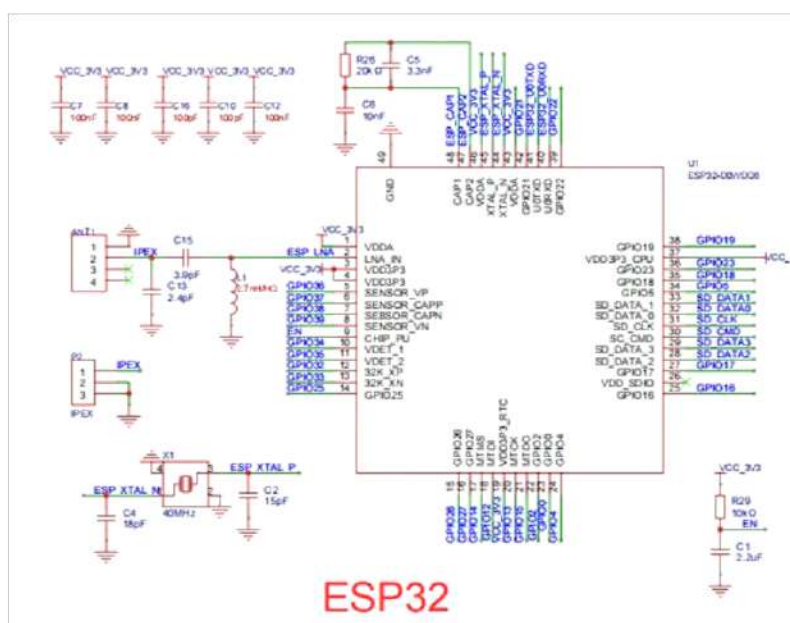


Рисунок 2.4 – Схема підключення компонентів мікроконтролера ESP32 [13]

2.1.2 Характеристики камери OV5640

Камера OV5640 є високопродуктивним CMOS сенсором із цифровим виходом, розробленим компанією OmniVision Technologies (рис. 2.5).



Рисунок 2.5 – Камера OV5640 з 66-градусним кутом огляду

Вона належить до класу матриць з технологією OmniBSI, що забезпечує покращену світлочутливість, низький рівень шумів і мінімальний перехресний вплив пікселів. Завдяки компактному форм-фактору та підтримці розширеного функціоналу, OV5640 активно застосовується в мобільних пристроях, системах відеоспостереження, мультимедійних платформах, а також у розробках на базі мікроконтролерів ESP32.

У модулі OV5640 реалізовано кілька апаратних алгоритмів автоматизації: автоматична експозиція, автоматичне підсилення, автоматичний баланс білого, фільтрація шумів, а також автофокус із вбудованим драйвером приводу VCM. Також сенсор підтримує корекцію лінзи, компенсацію битих пікселів і цифрові ефекти. Сенсор побудований на активному піксельному масиві 2592×1944 , що відповідає формату QSXGA. Кожен піксель має розмір 1.4×1.4 мкм, що дозволяє досягати високої щільності та деталізації зображення [15].

Архітектура роботи камери побудована наступним чином, ядро сенсора OV5640 формує потокові піксельні дані з постійною частотою кадрів, яка синхронізується сигналами HREF та VSYNC.

2.2 Аналіз протоколів передачі даних LoRaWAN, NB-IoT, MQTT

У межах розробки систем дистанційного зчитування показників лічильників на основі IoT важливим елементом є вибір протоколу передачі даних. Саме від обраної технології залежить енергоефективність, масштабованість, стабільність зв'язку та загальна надійність системи.

Найпоширенішими і найефективнішими на сьогодні вважаються протоколи LoRaWAN, NB-IoT, MQTT та HTTP – кожен із них має свої переваги й недоліки в контексті Інтернет речей.

2.2.1 Поняття та архітектура мережі LoRa

LoRaWAN (Long Range Wide Area Network) – це протокол, розроблений для забезпечення зв'язку між кінцевими пристроями LoRa. Він розшифровується як глобальна мережа великого радіусу дії та призначений для бездротових технологій у міжнародній або регіональній мережі. Хоча він має широку зону зв'язку, його енергоспоживання низьке.

Популярними сферами застосування LoRaWAN можна назвати системи безпеки, інтелектуальні вимірювання, розумні міста, промисловий контроль і технології автоматизації. За допомогою LoRaWAN можна досягати дальності зв'язку до 15 км. Для цього існує адаптивний діапазон передачі даних. Ця швидкість збалансована між дальністю зв'язку та часом передачі повідомлень. Мережеві з'єднання, такі як Ethernet, 3G або Wi-Fi, які втрачають свою функцію за відсутності електрики, зв'язуються через LoRaWAN, перш ніж передавати повідомлення за допомогою традиційних протоколів [16].

Кінцевий IoT-пристрій в мережі LoRaWAN такий як датчик, виконавчий механізм або обидва, підключається до бездротової мережі LoRaWAN через шлюзи модуляції LoRa. При цьому передача даних йде не на одну, а на всі наявні в даному місці базові станції. Відсутність прив'язки до конкретного шлюзу дає можливість гарантувати передачу інформації і при необхідності контролювати датчик в русі [16].

Уся інформація, що передається від кінцевої точки IoT, захищена наскрізним шифруванням із 2 рівнями криптографічного захисту, 128-бітним мережевим ключем і 128-бітним сеансовим ключем. Кожен шлюз зареєстрований у мережі LoRaWAN і надсилає отримані пакети даних безпосередньо на мережевий сервер, використовуючи стабільне мережеве підключення.

Мережевий сервер керує всією мережею, він отримує дані від шлюзів, видаляє дублікати повідомлень, надсилає отримані дані на відповідний сервер додатків та контролює швидкість передачі даних. Сервер додатків обробляє отримані дані та надсилає їх на підключені кінцеві пристрої. Дані можна інтерпретувати та використовувати для вирішення бізнес-проблем.

Серед головних переваг цієї технології варто можна виділити:

- винятково велика дальність радіозв'язку;
- висока здатність проникнення сигналу в забудованих або підземних умовах;
- простота впровадження завдяки топології типу «зірка»;
- тривалий час автономної роботи;
- відносно невисока вартість обладнання у порівнянні з іншими мережевими технологіями.

Ключові недоліки у використанні технології:

- затримка у передачі повідомлень;
- низька швидкість обміну.

2.2.2 Поняття та архітектура мережі NB-IoT

NB-IoT (Narrowband Internet of Things) – це стандарт зв'язку вузькосмугового типу, спеціально розроблений для підключення великої кількості пристроїв IoT, які працюють з мінімальним енергоспоживанням та передають невеликі обсяги даних. Він функціонує в ліцензованому спектрі мобільних операторів, забезпечуючи стабільний зв'язок навіть у важкодоступних місцях. Технологія NB-IoT дозволяє ефективно масштабувати

мережу до мільйонів пристроїв, зберігаючи при цьому високу автономність і надійність [17].

Можна виділити три ключові елементи архітектури: кінцеві пристрої (датчики, модулі), радіоступ та опорна мережа, яка відповідає за передачу, маршрутизацію та обробку даних.

Кінцеві пристрої NB-IoT, датчики, лічильники, контролери, які генерують і передають дані, зазвичай мають мінімальну обчислювальну потужність і працюють на батарейках, їх завдання фіксувати певні параметри середовища, вимірювати показники або реагувати на зміну умов. Ці пристрої надсилають дані через базові станції мобільного зв'язку, які використовують виділений або вбудований у LTE канал шириною лише 180 кГц.

Базові станції працюють як шлюзи, передаючи дані до основної мережі оператора, де відбувається автентифікація, обробка та перенаправлення інформації у хмарні платформи або сервери клієнта. Мережа підтримує як підключення пристроїв через інтернет, так і через приватні канали зв'язку, що є ефективно для промислових мереж.

Опорна мережа у NB-IoT відповідає за керування підключеннями, маршрутизацію трафіку та забезпечення безпеки даних. Вона включає компоненти, які обробляють автентифікацію пристроїв, управління сесіями, моніторинг трафіку та взаємодію з зовнішніми хмарними або прикладними сервісами.

Найважливішою особливістю NB-IoT є те, як вона оптимізована під автономність. Завдяки режимам збереження енергії, таким як PSM (режим сну з економією живлення) та eDRX (розширене переривчасте приймання), пристрої можуть залишатись у «сплячому режимі» більшість часу, активуючись лише для передачі чи прийому даних. Це дозволяє суттєво зменшити енергоспоживання, що критично важливо для зчитування показників лічильників [17].

Ще однією особливістю NB-IoT є її здатність до глибокого покриття, коли сигнал проникає крізь бетонні конструкції, досягає підвальних приміщень, каналізаційних колодязів або інших важкодоступних локацій. Тому NB-IoT

активно використовується в умовах міської забудови, на інфраструктурних об'єктах, у сільському господарстві та на виробництві (рис. 2.6)

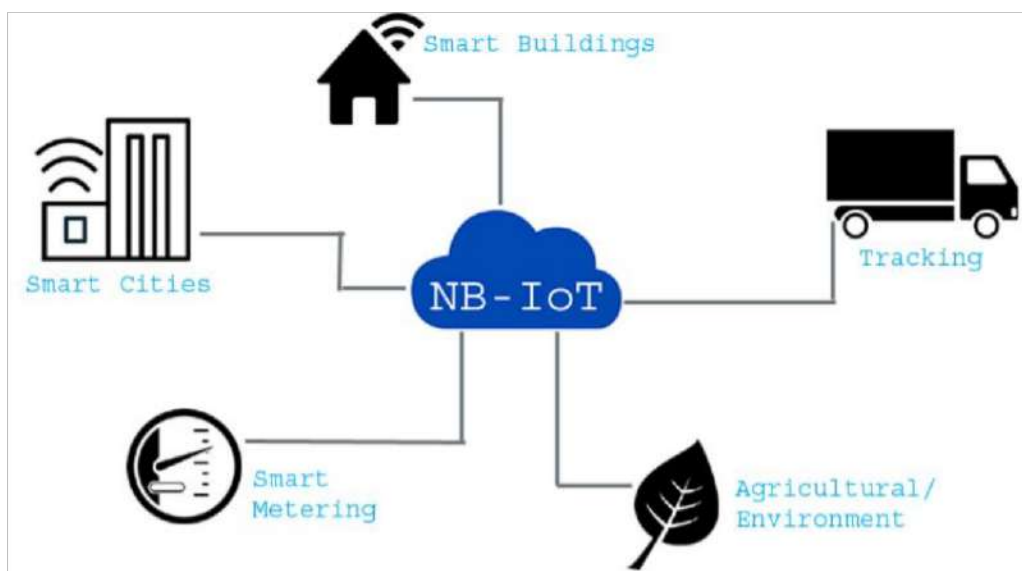


Рисунок 2.6 – Використання мережі NB-IoT [17]

Переваги NB-IoT:

- висока надійність передачі даних у важкодоступних місцях;
- підтримка масштабування до мільйонів пристроїв;
- низька вартість пристроїв і обслуговування;
- захищеність завдяки використанню мобільної інфраструктури.

Недоліки NB-IoT:

- залежність від мобільних операторів;
- низька швидкість обміну;
- може мати обмеження на підтримку двосторонньої передачі у певних реалізаціях.

2.2.3 Поняття та архітектура мережі MQTT

MQTT (Message Queuing Telemetry Transport) – це спеціалізований мережевий протокол, створений для передачі даних між пристроями з мінімальним навантаженням на мережу. Його розроблено таким чином, щоб він міг працювати в середовищах з обмеженим ресурсом, низькою пропускнуою здатністю, нестабільним підключенням та мінімальним енергоспоживанням [18].

У протоколі використовується модель зв'язку публікація-підписка, на відміну від традиційної моделі «запит-відповідь», яку реалізують HTTP чи REST API. Це означає, що пристрої, які генерують дані, не передають інформацію напряму до кожного отримувача. Замість цього вони публікують повідомлення у певні «теми», а ті пристрої, які підписані на ці теми, автоматично отримують відповідну інформацію [18].

MQTT-брокер – ключовий елемент системи, який забезпечує зв'язок, який керує чергами повідомлень, підписками на теми та кожним MQTT-з'єднанням, забезпечуючи безперебійний зв'язок пристрою. Брокер приймає повідомлення від видавців, обробляє їх і надсилає відповідним підписникам, які підписані на певні теми. Він також виконує автентифікацію та авторизацію клієнтів, забезпечуючи безпечне та ефективне користування в мережі MQTT.

Теми в MQTT – це шлях за яким брокер ідентифікує, які дані надійшли і куди їх доставити. Теми мають ієрархічну структуру, подібну до каталогів у файловій системі. Це дозволяє логічно розділяти повідомлення від різних пристроїв і гнучко організовувати обробку даних.

Клієнт MQTT – це будь-який пристрій або програма, що може надсилати чи отримувати повідомлення. Клієнти можуть працювати на різних мовах програмування та операційних системах, і легко інтегруються з контролерами Arduino, ESP32, Raspberry Pi та іншими.

Переваги MQTT:

- мінімальне споживання мережевих ресурсів;
- підтримка нестабільних з'єднань;
- висока швидкість доставки повідомлень;
- підтримка різних рівнів якості обслуговування QoS;
- простота інтеграції з різними платформами та мовами.

Недоліки MQTT:

- потребує постійного з'єднання з брокером;
- не придатний для великих обсягів даних;
- вимагає ручного керування автентифікацією та шифруванням.

2.2.4 Поняття та архітектура протоколу HTTP

HTTP (HyperText Transfer Protocol) – це протокол прикладного рівня, який використовується для передачі гіпертекстових документів у мережі Інтернет. Його основна функція полягає у забезпеченні взаємодії клієнта та сервера для обміну інформацією у форматі веб-сторінок, файлів, зображень та інших ресурсів. HTTP працює за моделлю «запит-відповідь», де клієнт надсилає запит на сервер, а сервер повертає відповідь із запитаним ресурсом або повідомленням про помилку. Це основний протокол для роботи всесвітньої павутини і він стандартизований в рамках IETF у RFC 2616 та його наступниках [19].

Архітектура HTTP базується на кількох ключових компонентах, а саме, клієнт, сервер, протокол передачі і набір методів для управління запитами. HTTP використовує порт 80 для незашифрованого трафіку та 443 для захищеного, де останній є розширенням протоколу з доданим шифруванням через TLS/SSL.

Клієнт HTTP – це програмне забезпечення, яке формує запити і відправляє їх серверу. Це може бути веб-браузер, мобільний додаток або спеціалізований клієнтський модуль. Запит містить інформацію про метод, URL ресурс, заголовки та іноді тіло запиту. Методи визначають дію, яку клієнт хоче виконати з ресурсом, отримати, створити, оновити або видалити дані.

Сервер HTTP приймає запити, обробляє їх та формує відповідь, яка складається з статусного коду, наприклад, 200 – ОК, 404 – Not Found, заголовків і, за потреби, тіла з даними, наприклад, HTML, JSON, зображення. Сервери можуть бути як простими статичними, так і складними додатками, що обробляють бізнес-логіку і взаємодіють з базами даних.

HTTP є безстанним протоколом, що означає, що сервер не зберігає інформацію про попередні взаємодії, тому для підтримки користувачьких сесій використовуються додаткові механізми, такі як куки або токени авторизації [19]. Архітектура HTTP базується на мережевій моделі, де протокол прикладного рівня взаємодіє з транспортним та мережевим рівнями. На транспортному рівні працює протокол TCP (Transmission Control Protocol), який забезпечує надійну, орієнтовану на з'єднання передачу даних. TCP відповідає за встановлення

з'єднання між клієнтом і сервером, контроль цілісності та порядок доставки пакетів. На мережевому рівні функціонує протокол IP (Internet Protocol), який відповідає за маршрутизацію пакетів даних у глобальній мережі. IP визначає адресацію пристроїв та передає пакети від джерела до призначення, незалежно від кількості проміжних вузлів.

HTTP використовує TCP/IP стек, що забезпечує поєднання надійності та масштабованості передачі даних. TCP створює стабільний канал для HTTP-запитів і відповідей, а IP відповідає за їх доставку у великій мережі Інтернет.

HTTP за замовчуванням є безстанним протоколом, тобто не зберігає інформацію про попередні взаємодії, що спрощує його реалізацію, але вимагає додаткових механізмів для роботи з сесіями, таких як куки або токени.

Для забезпечення безпеки передачі HTTP може працювати поверх TLS/SSL, що реалізує протокол HTTPS, захищену версію HTTP із шифруванням.

Переваги HTTP:

- простий і зрозумілий протокол з моделлю «запит-відповідь»;
- широко використовується і підтримується всіма браузерами та серверами;
- підтримує захищене з'єднання через HTTPS;
- гнучкий завдяки різним методам та заголовкам;
- простота реалізації та використання.

Недоліки HTTP:

- відносно висока затримка через часте відкриття з'єднань (HTTP/1.1);
- безстанність протоколу ускладнює роботу з сесіями;
- відсутність захисту в базовій версії без HTTPS.

2.3 Особливості використання програмних технологій

Для програмування контролера використовувалось середовище розробки Arduino IDE.

Arduino IDE – це безкоштовне інтегроване середовище розробки, призначене для створення, компіляції та завантаження програмного коду на мікроконтролери Arduino. Це середовище підтримує операційні системи Windows, macOS та Linux і надає користувачам інструменти для написання, перевірки та завантаження коду на плату Arduino.

Самої індивідуальної мови, як і компілятора даного середовища розробки немає тому програми, що створюються в Arduino IDE, називаються скетчами та пишуться на спрощеній версії мови програмування C/C++. Сам код компілюється за допомогою компілятора AVR-GCC або іншого йому відповідного, залежно від архітектури мікроконтролера. Після компіляції код завантажується на мікроконтролер через USB-з'єднання. Усі процеси від компіляції до завантаження прошивки на пристрій здійснюються в межах самого середовища, що значно спрощує розробку [20].

Інтерфейс Arduino IDE складається з простого редактора коду, повідомлень компілятора, кнопок управління завантаженням, перевіркою коду та вибором плати або порту. Середовище також містить вбудовану бібліотеку функцій, які значно спрощують роботу з периферією, цифровими та аналоговими входами. За потреби можна підключати сторонні бібліотеки, завантажувати нові драйвери плат або додавати підтримку нестандартних архітектур, таких як, наприклад, ESP32 [20].

Arduino IDE також підтримує створення інтерфейсів для роботи з різними модулями та датчиками, завдяки спрощеному процесу налаштування, користувачі можуть легко підключати різноманітні периферійні пристрої, такі як датчики температури, вологості, світла або навіть керування моторами. Вбудовані функції для роботи з цифровими та аналоговими входами/виходами значно зменшують час на розробку. Крім того, середовище підтримує створення власних бібліотек для специфічних завдань, що робить його універсальним для широкого спектра проектів.

Це середовище розробки ідеально підходить як для новачків, так і для досвідчених розробників, оскільки поєднує в собі простоту використання з потужними можливостями для створення складних проектів.

Нижче на рисунку 2.7 представлено зображення середовища розробки Arduino.

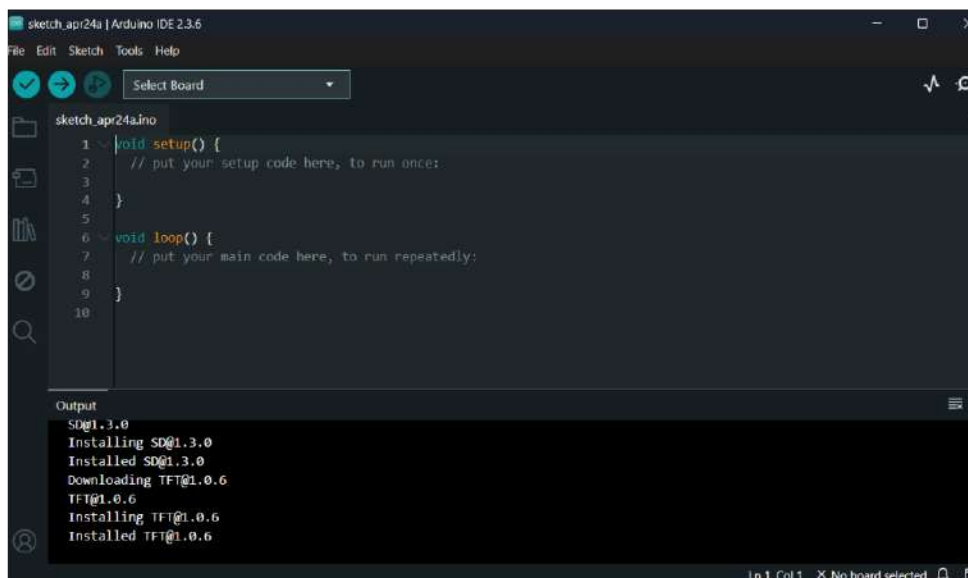


Рисунок 2.7 – Середовище розробки програми

Arduino IDE та переваги:

- доступність;
- зручність у використанні інтерфейсу;
- сумісність з усіма операційними системами Windows;
- наявність базових інструментів для роботи;
- різноманіття мов програмування;
- вбудований широкий вибір прикладів програм;
- працює з різними платами Arduino та сумісними мікроконтролерами.

Arduino IDE та недоліки:

- програма не ефективно справляється з складними проектами;
- окремі версії характеризуються нестабільністю;
- повільна компіляція.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРИСТРОЮ

3.1 Налаштування апаратного та програмного забезпечення

Налаштування апаратного забезпечення є важливим етапом у реалізації системи дистанційного зчитування показників лічильників з використанням технологій IoT на базі ESP32-CAM з камерою OV5640. Коректна конфігурація пристрою гарантує стабільність роботи, забезпечує взаємодію між компонентами та визначає успішність виконання всього проекту.

На першому етапі налаштування необхідно правильно зібрати апаратну схему. Для цього плата ESP32-CAM повинна бути з'єднана з камерою OV5640. Важливо уважно дотримуватись інструкцій, наведених у технічній документації, для уникнення можливих помилок підключення. Камера під'єднується за допомогою гнучкого шлейфу до спеціального роз'єму на платі ESP32-CAM. Цей роз'єм розташований на верхній частині плати та має чітке позначення. Важливо переконатись, що шлейф правильно і щільно вставлени, щоб запобігти перебоям у роботі пристрою.

Після успішного збирання апаратної частини необхідно здійснити налаштування середовища Arduino IDE, оскільки саме воно є основним програмним інструментом для завантаження та налагодження програмного забезпечення пристрою.

Спершу необхідно завантажити Arduino IDE з офіційного сайту та встановити на комп'ютер з підтримкою відповідної операційної системи.

Після встановлення Arduino IDE необхідно додати підтримку плати ESP32. Для цього у вікні налаштувань середовища Arduino IDE слід додати посилання на офіційний пакет підтримки ESP32 від Espressif Systems, це посилання додається в полі «Додаткові URL-адреси менеджера плат» (рис. 3.1).

Після додавання посилання необхідно відкрити менеджер плат, знайти модуль ESP32 та встановити його. У процесі встановлення, середовище автоматично завантажить усі бібліотеки та драйвери, для коректної роботи плат.

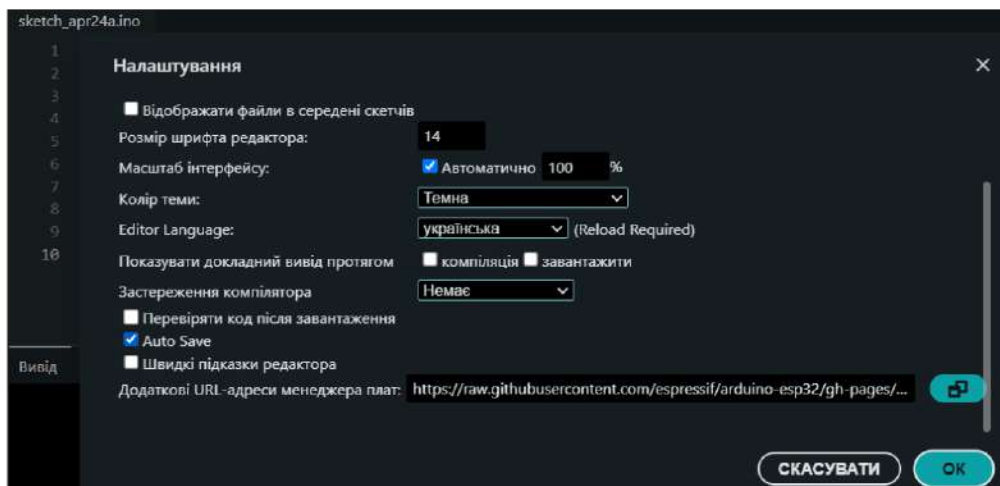


Рисунок 3.1 – Налаштування підтримки ESP32

Після цього потрібно перейти до менеджера плат Arduino IDE і знайти за допомогою пошуку пакет «esp32» від Espressif Systems. Встановивши цей пакет, користувач отримує доступ до необхідних бібліотек та налаштувань для роботи з платами ESP32 (рис. 3.2).

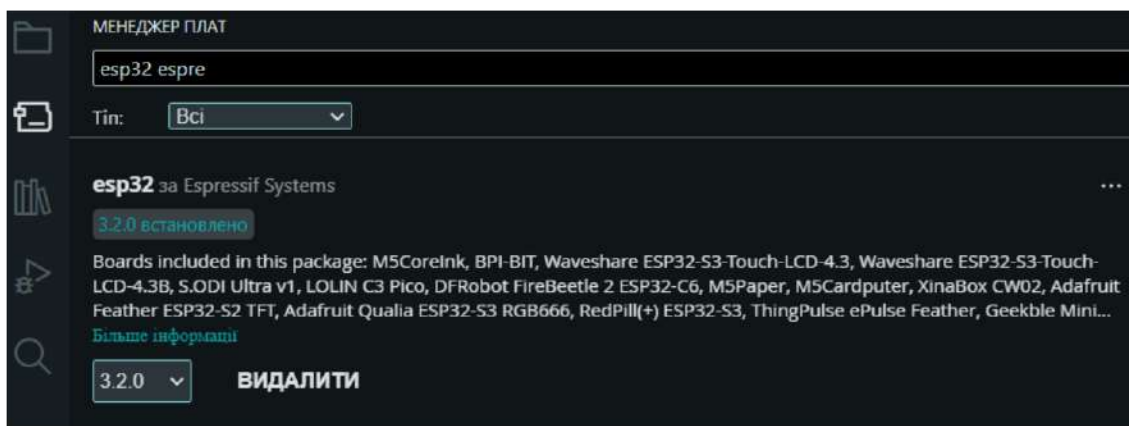


Рисунок 3.2 – Встановлення бібліотеки з платами ESP32

Наступним важливим кроком є конфігурація параметрів плати у середовищі Arduino IDE. У меню вибору плати слід встановити «ESP32-CAM», після чого вибрати оптимальну частоту процесора, тип та розмір пам'яті, а також порт підключення. Вибір цих параметрів має суттєве значення, оскільки невідповідність налаштувань може спричинити помилки при завантаженні програми або неправильну роботу пристрою (рис. 3.3).

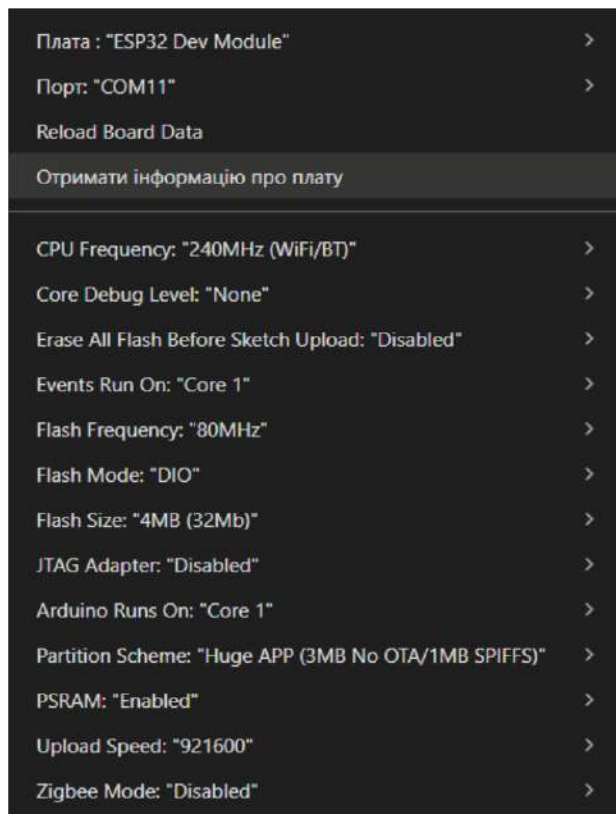


Рисунок 3.3 – Налаштування плати ESP32 у середовищі Arduino IDE

Обрана плата «ESP32 Dev Module» забезпечує сумісність з мікроконтролером, а порт COM11 використовується для підключення пристрою до комп'ютера під час програмування.

Частота роботи процесора встановлена на максимальні 240 МГц, що дозволяє обробляти великі обсяги даних, зокрема зображення, а також підтримує стабільне функціонування Wi-Fi та Bluetooth модулів. Рівень відладки відсутній, що сприяє мінімальному навантаженню під час роботи. При завантаженні нової прошивки видалення всіх даних із флеш-пам'яті вимкнене, що дозволяє зберегти налаштування між оновленнями.

Операції з пам'яттю флеш-пам'яті виконуються на частоті 80 МГц у режимі DIO, який забезпечує швидкий обмін даними. Загальний обсяг флеш-пам'яті складає 4 МБ, з яких 3 МБ виділено під основний додаток, а 1 МБ під файлову систему SPIFFS, необхідну для зберігання додаткових даних. Використання зовнішньої оперативної пам'яті PSRAM значно підвищує

продуктивність при обробці зображень з камери OV5640, що є важливим для проекту.

Швидкість передачі коду на плату встановлена на 921600 бод, що дозволяє прискорити процес програмування.

Режим Zigbee вимкнено, оскільки він не використовується в цьому застосуванні.

Загалом ці налаштування забезпечують оптимальний баланс між продуктивністю, стабільністю роботи та обсягом доступних ресурсів, що є необхідним для успішного функціонування мікроконтролера.

Головним завданням коду є забезпечення роботи камери, отримання зображень лічильника та їх передача через бездротову мережу Wi-Fi на сервер для подальшої обробки. Для цього в коді реалізовані декілька основних блоків логіки: ініціалізація та налаштування апаратури, організація мережевого з'єднання, робота веб-сервера, зйомка зображень та їх трансляція, а також механізми дистанційного налаштування параметрів камери.

Першим завданням було забезпечити ініціалізацію камери OV5640, яка виконується через стандартний API ESP-IDF, адаптований у бібліотеках Arduino для ESP32. У цьому етапі відбувається налаштування параметрів камери, таких як, вибір формату зображення, розміру кадру, наприклад, VGA, SVGA, якості стиснення, яскравості, контрасту, насиченості кольорів та інших характеристик. Ці налаштування дозволяють адаптувати пристрій під умови освітлення та вимоги до чіткості знімків. У коді передбачені функції для зміни цих параметрів під час роботи через веб-інтерфейс, що забезпечує гнучкість і зручність керування пристроєм без необхідності перепрошивки.

Другий ключовий крок, це налаштування мережі Wi-Fi, код містить логіку підключення ESP32 до заданої Wi-Fi мережі, отримання IP-адреси і запуск веб-сервера на цій адресі. Завдяки цьому до пристрою можна підключитися будь-якому клієнту через браузер або спеціальний додаток, що підтримує протокол HTTP.

Веб-сервер у кодї реалізований за допомогою HTTP серверної бібліотеки ESP-IDF, яка дозволяє обробляти різні HTTP-запити. Для кожного типу запиту створені окремі обробники:

- головна сторінка, яка містить веб-інтерфейс для віддаленого перегляду відео та налаштування параметрів камери;
- запити для отримання окремих знімків у форматі JPEG або BMP;
- запити для отримання потоку MJPEG, що дозволяє в режимі реального часу транслювати відео з камери;
- запити, що змінюють параметри камери, наприклад, розмір кадру, якість зображення, баланс білого тощо.

Реалізація потокового відео потребувала організації ефективної обробки кадрів кожен кадр захоплюється камерою, за потреби конвертується в JPEG, потім надсилається у вигляді частин HTTP-відповіді із встановленими межами. Такий підхід дозволяє браузеру відображати відео як послідовність JPEG-зображень.

Важливою частиною коду є механізм читання параметрів з URL-запитів та їх обробка. Це дозволяє змінювати налаштування камери в режимі реального часу, просто відправляючи HTTP-запити з відповідними параметрами. Наприклад, зміна розміру кадру, якості стиснення, яскравості, контрасту та інших властивостей відбувається через такі запити, що значно підвищує гнучкість системи.

Для підвищення стабільності відеопотоку в кодї використовується фільтр ковзного середнього для часу між кадрами, що дозволяє відстежувати і згладжувати коливання частоти кадрів, забезпечуючи плавність відображення.

3.2 Схема підключення та алгоритм роботи пристрою

Схема підключення пристрою складається з мікроконтролера ESP32-CAM з камерою OV5640, що виконує функції зчитування зображень з лічильника. Основним джерелом живлення є стабілізований блок живлення, що забезпечує

напругу 5 В з рекомендованим струмом до 2 А. Джерелом освітлення є світлодіод припаяний до плати, а мікроконтролер через вбудований Wi-Fi модуль підключений до локальної мережі, в межах якої забезпечується обмін даним з сервером. Камера за допомогою спеціалізованого шлейфу з'єднана з мікроконтролером, забезпечуючи стабільну передачу зображень. Нижче наведено схему, що пояснює принцип роботи пристрою (рис. 3.4).

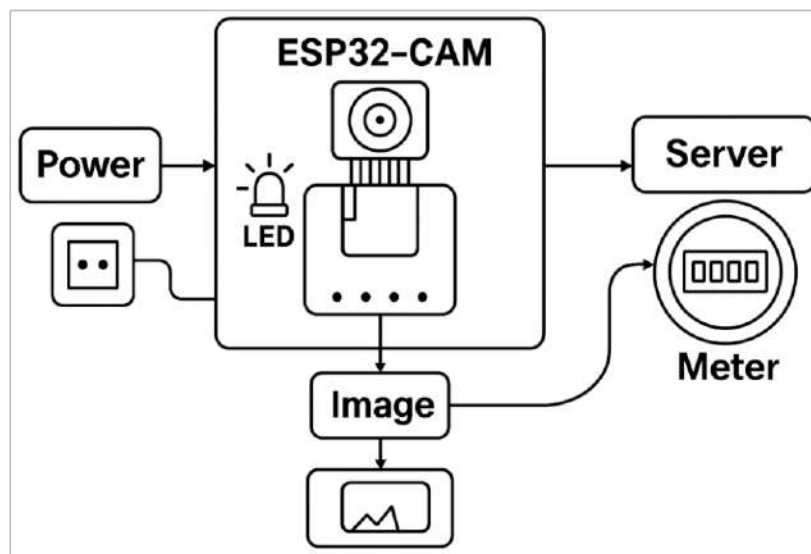


Рисунок 3.4 – Принцип роботи пристрою

Після подачі живлення на пристрій починається послідовний процес роботи, який забезпечує безперервний моніторинг показників, спочатку мікроконтролер виконує ініціалізацію всіх необхідних модулів таких як, камера, Wi-Fi інтерфейс, пам'ять. Ініціалізація передбачає налаштування параметрів камери для отримання оптимального зображення, а також підключення до заданої локальної Wi-Fi мережі, де пристрій отримує унікальну IP-адресу. Ця адреса є ключовою для подальшої взаємодії з пристроєм через веб-браузер.

Далі починається періодичний цикл зчитування даних, при взаємодії з камерою, робиться фото циферблата лічильника, створюючи цифрове зображення з оптимальною роздільною здатністю та освітленням за рахунок світлодіода. Висока якість отриманих знімків є необхідною для точного розпізнавання цифр у подальшому аналізі зображення.

Наступним кроком відбувається передача отриманого зображення на сервер у межах локальної мережі. Для цього використовується протокол HTTP, який забезпечує надійний обмін даними між пристроєм та сервером. Сервер, отримуючи зображення, виконує їх збереження та подальшу обробку.

Такий підхід дозволяє мінімізувати навантаження на мікроконтролер та підвищує гнучкість системи, оскільки обробка може відбуватися на більш потужному серверному обладнанні.

На рисунку 3.5 проілюстровано ключову блок-схему алгоритму роботи пристрою.

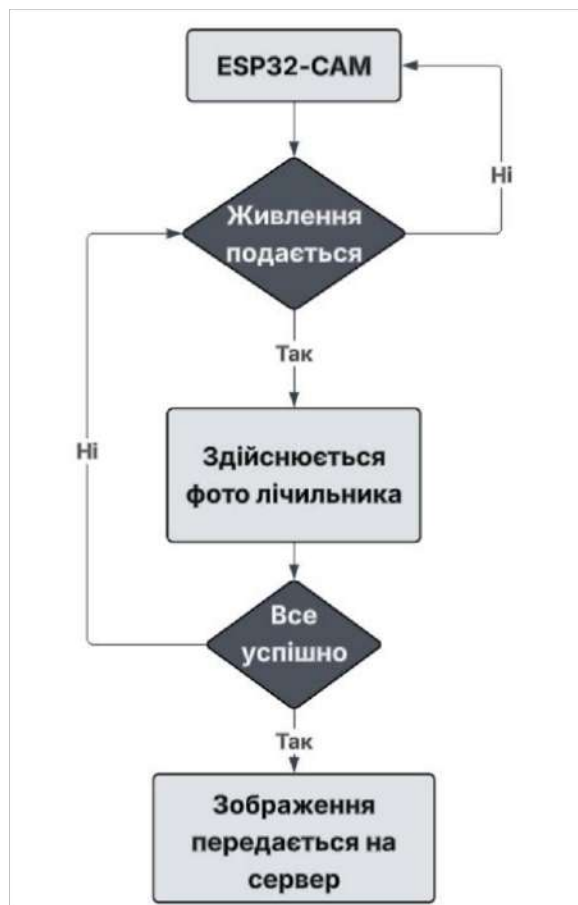


Рисунок 3.5 – Блок-схема алгоритму роботи пристрою

3.3 Тестування та оптимізація пристрою

Після завершення етапу програмування та апаратного налаштування було проведено тестування системи дистанційного зчитування показників

лічильників з метою оцінки якості зображення, стабільності роботи та підбору оптимальних параметрів камери для точного розпізнавання цифр. Зокрема, було протестовано параметри, які безпосередньо впливають на якість зображення, роздільна здатність, яскравість, контраст, насиченість кольорів, різкість, рівень експозиції та активність автоматичних режимів обробки зображення.

Початковий вигляд інтерфейсу та зображення з камери (рис. 3.6) показували, що при стандартних параметрах яскравість і контрастність були недостатні для комфортного зчитування показників. На темному фоні важко було ідентифікувати всі цифри, а рівень шумів був вищим, ніж очікувалось.

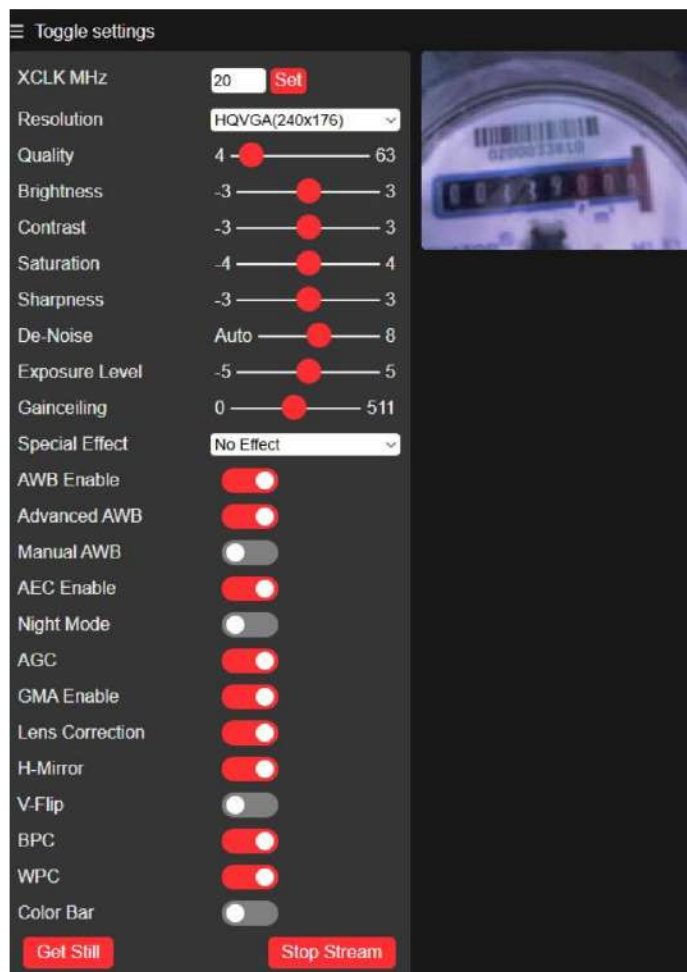


Рисунок 3.6 – Початковий вигляд інтерфейсу та зображення з камери

На момент першого тесту камера працювала з активованими режимами автоматичного балансу білого, розширеного AWB, автоматичного контролю експозиції, функцією вертикального віддзеркалення та активним кольоровим

тестовим шаблоном. Якість була встановлена на значенні 12, а роздільна здатність HQVGA забезпечувала достатньо швидке оновлення кадру з мінімальним навантаженням, проте значення яскравості, контрасту, насиченості та експозиції потребували налаштування.

Однак, деякі налаштування, такі як яскравість, контраст, насиченість кольорів та експозиція, вимагали додаткового налаштування для досягнення більш чіткого та природного зображення.

У результаті серії експериментів було сформовано фінальний варіант конфігурації камери, який продемонстрував найкращу якість зображення (рис. 3.7).

У цьому режимі було внесено кілька ключових змін у налаштування, які суттєво вплинули на покращення візуального результату, а саме, значення контрасту було збільшено з 0 до +1, що підсилило різницю між темними і світлими зонами та зробило межі цифр чіткішими, параметр насиченості кольорів також було підвищено з 0 до +1, що усунуло блідість і повернуло зображенню природний вигляд, зміна яскравості з 0 до -2 допомогла приглушити надмірно освітлені ділянки, зокрема при спробах зйомки з лічильників, які мають світлові відблиски.

Окрему увагу було приділено параметру якість зображення, який визначає ступінь стиснення JPEG-знімків. У початковій конфігурації це значення було встановлено на рівні 10, що відповідало середньому рівню стискання та зменшувало деталізацію через компресію, після тестів було прийнято рішення зменшити це значення до 4, тобто суттєво підвищити якість зображення шляхом зменшення ступеня стискання.

Проведені налаштування дозволили передавати більш деталізовані кадри з кращою чіткістю, особливо в зонах, де на лічильнику знаходяться тонкі лінії та межі цифр, зображення покращилось навіть у умовах з різними рівнями освітлення та відображеннями на лічильниках. Це стало важливим етапом у забезпеченні надійності системи, оскільки саме від якості зображення залежить точність подальшого зчитування показників.



Рисунок 3.7 – Фінальний вигляд інтерфейсу та зображення з камери

Після застосування оновлених налаштувань на зображенні було досягнуто помітного підвищення контрастності та загальної читабельності, цифри стали чіткішими, фон рівномірнішим, а самі межі символів краще вираженими (рис. 3.8). Такий результат показав, що саме комбіноване коригування рівня стискання зображення, контрастності та усунення автоматичних режимів може дати найкращу якість без зайвих ресурсних витрат.



Рисунок 3.8 – Зображення з камери

Отримане зображення є не лише якісним для візуального спостереження, але й повністю придатним для подальшої цифрової обробки, зокрема, завдяки достатній чіткості контурів та стабільному контрасту, зображення можна ефективно використовувати у поєднанні з сучасними технологіями комп'ютерного зору.

Одним із ключових напрямків є застосування алгоритмів оптичного розпізнавання символів, які дозволяють автоматично зчитувати цифрові значення з лічильників. Це відкриває можливості для повної автоматизації процесу збору показників без участі людини, надалі такі рішення можуть бути інтегровані у більші системи моніторингу або обліку споживання ресурсів, що дає змогу зменшити витрати часу, уникнути людських помилок та підвищити ефективність обслуговування у сфері комунального господарства чи промисловості.

На рисунку 3.9 зображено зовнішній вигляд системи дистанційного зчитування показників лічильників на основі іот у парі з засобом обліку холодної води.



Рисунок 3.9 – Зовнішній вигляд пристрою

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було запропоновано сучасне та ефективне рішення для проблеми дистанційного зчитування показників лічильників на основі технології Інтернету речей. Проведено детальний аналіз існуючих систем, протоколів передачі даних і методів автоматизації, що дозволило обґрунтувати вибір апаратної платформи ESP32-CAM із камерою OV5640 та бездротових протоколів Wi-Fi і HTTP для реалізації надійної системи збору даних.

Було спроектовано і реалізовано повноцінну систему збору та передачі показників лічильників, яка включає в себе апаратний пристрій для зчитування інформації, мережевий інтерфейс для передачі даних і серверну частину для їх обробки та зберігання. Пристрій на базі ESP32-CAM із вбудованою камерою OV5640 дозволяє автоматично фіксувати зображення циферблата лічильника, що усуває необхідність ручного зчитування і зменшує ризик помилок.

Розроблена серверна система забезпечує надійне приймання, збереження та аналіз зібраних даних, а також має можливість віддаленого налаштування параметрів пристрою через веб-інтерфейс. Це робить систему гнучкою, масштабованою та зручною у використанні для різних категорій користувачів, від комунальних підприємств до приватних споживачів.

Проведено дослідження ефективності обраних компонентів, оцінено енергоспоживання, стабільність роботи апаратної частини та якість зображень, отриманих камерою OV5640. Також здійснено порівняння різних мережевих протоколів, таких як LoRaWAN, NB-IoT, MQTT і HTTP, за критеріями енергоефективності, стабільності та дальності передачі даних. Результати досліджень підтвердили високу продуктивність і надійність запропонованої системи, що дозволяє впроваджувати її у реальних умовах з урахуванням економічної доцільності.

Таким чином, розроблену систему можна рекомендувати для використання у сфері дистанційного обліку води, газу, електроенергії та інших ресурсів.

Впровадження такої системи сприятиме підвищенню точності збору даних, зниженню операційних витрат, покращенню контролю за споживанням і підвищенню рівня автоматизації в комунальному господарстві.

У перспективі можливе розширення функціональності системи за рахунок інтеграції алгоритмів машинного навчання для автоматичного розпізнавання та аналізу зображень, а також підтримки додаткових типів лічильників та протоколів зв'язку. Це дозволить створити більш комплексну платформу для моніторингу ресурсів у реальному часі з високою точністю і надійністю.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Церпіш І., Костючко С., Конкевич Л. Система дистанційного зчитування показників лічильників на основі іот. *Програмне та апаратне забезпечення в інформаційних технологіях* : зб. тез доп. міжнар. наук.-практ. конф. молодих науковців та студентів, м. Луцьк, 6 травня 2025 р. / Луцький національний технічний університет. Луцьк, 2025. С. 180-182.
2. Засорнов О. С., Засорнова І. О. Програмування мікроконтролерних та робототехнічних систем. Кондор, 2023. 280 с.
3. An overview of IoT architectures, technologies, and existing open-source projects / T. Domínguez-Bolaño et al. *Internet of Things*. 2022. P. 100626. URL: <https://doi.org/10.1016/j.iot.2022.100626> (дата звернення: 12.02.2025).
4. Optimizing IoT Energy Efficiency: Real-Time Adaptive Algorithms for Smart Meters with LoRaWAN and NB-IoT / K. A. Al-Sammak et al. *Energies*. 2025. Vol. 18, no. 4. P. 987. URL: <https://doi.org/10.3390/en18040987> (дата звернення: 16.02.2025).
5. IoT-Based Energy Meter for Remote Monitoring and Managements of Power Consumption / A. A. Kadhim et al. *International Journal of Electrical and Electronics Research*. 2024. Vol.12, no. 4.P. 1427-1439. URL: <https://surl.li/nyyebx> (дата звернення: 24.02.2025).
6. Okoli N. J. Smart Water Metering System (SWMS) Adoption: A Systematic Literature Review. *International Conference on Artificial Intelligence and its Applications*. 2023. Vol. P. 172-180. URL: <https://doi.org/10.59200/icarti.2023.025> (дата звернення: 28.02.2025).
7. Knayer T., Kryvinska N. An analysis of smart meter technologies for efficient energy management in households and organizations. *Energy Reports*. 2022. Vol. 8. P. 4022–4040. URL: <https://doi.org/10.1016/j.egyr.2022.03.041> (дата звернення: 02.03.2025).
8. Prom – найбільший маркетплейс України. *prom.ua*. URL: <https://prom.ua/ua/> (дата звернення: 04.03.2025).

9. Дистанційний збір показників лічильників Передача даних і аналітика витрат. *Jooby*. URL: <https://jooby.eu/uk/rdc-landing-page-ua/> (дата звернення: 09.03.2025).
10. Системи на базі M-Bus. *Antap*. URL: <https://antap.com.ua/catalog/sistemi-na-bazi-m-bus> (дата звернення: 10.03.2025).
11. Smart Energy and Water Solutions – Itron. *Itron*. URL: <https://na.itron.com> (дата звернення: 12.03.2025).
12. The evolution of the Internet of Things (IoT) over the past 20 years / J. Wang et al. *Computers & Industrial Engineering*. 2021. Vol. 155. P. 107174. URL: <https://doi.org/10.1016/j.cie.2021.107174> (дата звернення: 13.03.2025).
13. Sathish J. Learn Esp32 With Arduino: Arduino Coding, ESP32 Coding, Circuit Diagram, IoT Projects, MQTT. Independently published, 2021. 214 p.
14. ESP32 Pdf. *alldatasheet.com - Electronic Parts Datasheet Search* URL: <https://surl.li/jftkaa> (дата звернення: 17.03.2025).
15. OV5640 Pdf. *alldatasheet.com - Electronic Parts Datasheet Search* URL: <https://surli.cc/nwrwxz> (дата звернення: 19.03.2025).
16. What is LoRaWAN Technology? What are the advantages? – Aktif Elektrotechnik. *Aktif Elektrotechnik*. URL: <https://aktif.net/en/what-is-lorawan-technology-what-are-the-advantages/> (дата звернення: 20.03.2025)
17. What is NB-IoT? - everything RF. *everything RF - The leading website for the RF & Microwave Industry*. URL: <https://www.everythingrf.com/community/what-is-nb-iot> (дата звернення: 25.03.2025).
18. MQTT protocol in iot: a guide to reliable iot communication. *cavliwireless.com*. URL: <https://www.cavliwireless.com/blog/nerdiest-of-things/what-is-the-mqtt-protocol> (дата звернення: 26.03.2025).
19. Wendroth J., Jaeger B. A Brief Overview on HTTP. *Network Architectures and Services*. 2022. Vol. 11, no. 1. P. 59-63. URL: <https://surl.li/tiducz> (дата звернення: 01.04.2025).
20. Parker D. Arduino Programming: The Ultimate Guide for Making the Best of Your Arduino Programming Projects. Independently Published, 2020. 245 p.

21. Кваліфікаційна робота Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія денної та заочної форм навчання/ уклад. С.В. Лавренчук, Н.В. Багнюк. Луцьк: ЛНТУ, 2025. 56 с.

ДОДАТКИ

Додаток А

Основний код програми

```
#include "esp_camera.h"
#include <WiFi.h>

#define RESET_GPIO_NUM 5
#define XCLK_GPIO_NUM 15
#define SIOD_GPIO_NUM 22
#define SIOC_GPIO_NUM 23

#define D0_GPIO_NUM 2
#define D1_GPIO_NUM 14
#define D2_GPIO_NUM 35
#define D3_GPIO_NUM 12
#define D4_GPIO_NUM 27
#define D5_GPIO_NUM 33
#define D6_GPIO_NUM 34
#define D7_GPIO_NUM 39

#define VSYNC_GPIO_NUM 18
#define HREF_GPIO_NUM 36
#define PCLK_GPIO_NUM 26

#define LED_GPIO_NUM 25

const char *ssid = "";
const char *password = "";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();
```

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = D0_GPIO_NUM;
config.pin_d1 = D1_GPIO_NUM;
config.pin_d2 = D2_GPIO_NUM;
config.pin_d3 = D3_GPIO_NUM;
config.pin_d4 = D4_GPIO_NUM;
config.pin_d5 = D5_GPIO_NUM;
config.pin_d6 = D6_GPIO_NUM;
config.pin_d7 = D7_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = -1;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

config.frame_size = FRAMESIZE_HQVGA;
config.jpeg_quality = 10;
config.fb_count = 1;

esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t *s = esp_camera_sensor_get();
```

```
pinMode(LED_GPIO_NUM, OUTPUT);
digitalWrite(LED_GPIO_NUM, HIGH);

WiFi.begin(ssid, password);
WiFi.setSleep(false);

Serial.print("WiFi connecting");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

#if defined(LED_GPIO_NUM)
    setupLedFlash(LED_GPIO_NUM);
#endif

startCameraServer();

Serial.print("Camera Ready! Use `http://");
Serial.print(WiFi.localIP());
Serial.println("` to connect");

}

void loop() {
    delay(10000);
}
```