

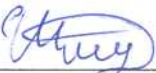
МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем

КВАЛІФІКАЦІЙНА РОБОТА

**на тему: КОМП'ЮТЕРИЗОВАНА СИСТЕМА ДИСТАНЦІЙНОГО
МОНІТОРИНГУ ЯКОСТІ ПИТНОЇ ВОДИ**

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-20 -1,
спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Шмультський Олександр
Олегович

Науковий керівник


(підпис)

Горбовий А.Ю ., д.т.н., професор

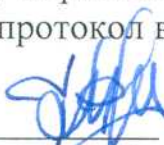
Керівник освітньо-професійної
програми


(підпис)

Омельчук А.А., к.т.н.

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем, протокол від «22» травня 2023 № 15

Завідувач кафедри


(підпис)

Філоненко М. М., к.ф.-м.н., доцент

Ірпінь – 2024

Реєстрація № 56 31.05.2024
(дата)


(підпис відповідальної особи)

Результати перевірки до захисту
(до захисту, на доопрацювання)

22.05.2024
(дата)


(підпис)

Результат захисту 3 (задовільно)
(оцінка)

12.06.2024
(підпис)


(дата)

Голова комісії:


(підпис)

В.М. Франчук
(ініціали, прізвище)

Члени комісії:


(підпис)

М.М. Філоненко
(ініціали, прізвище)


(підпис)

С.М. Грищенко


(підпис)

В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	18		12.06.2024 Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i> Грищенко С.М. <i>С.М.</i>
2. Дотримання вимог щодо оформлення	від 0 до 10	7		Ніжегородцев В.О. <i>В.О.</i>
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		Суховерцька Л. <i>Л.</i>
4. Рецензент	від 0 до 5	3		
РАЗОМ БАЛІВ	від 0 до 50	31		

Оцінювання кваліфікаційної роботи науковим керівником - 30 балів.

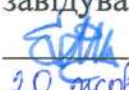
ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
61	D	3 (задовільно)	12.06.2024	Франчук В.М. <i>В.М.</i> Філоненко М.М. <i>М.М.</i> Грищенко С.М. <i>С.М.</i> Ніжегородцев В.О. <i>В.О.</i> Суховерцька Л. <i>Л.</i>

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	
70-79	C	4 (добре)
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні управляючі системи та технології (за галузями)»

Затверджую
завідувач кафедри
 Філоненко М. М.
20 жовтня 2023

Завдання на кваліфікаційну роботу
здобувача вищої освіти
Шмутьського Олександра Олеговича

1. Тема кваліфікаційної роботи: Комп'ютеризована система дистанційного моніторингу якості питної води
затверджена наказом ректора від 20.10. 2023 №1411
2. Строк здачі закінченої роботи: 20. травня 2024
3. Цільова установка: застосування набутих знань для розробки комп'ютеризованої системи дистанційного моніторингу якості питної води.
Мета роботи: вдосконалення комп'ютеризованої системи для дистанційного моніторингу якості питної води.
Об'єкт дослідження: якість питної води. засоби дистанційного моніторингу якості питної води.
Предмет дослідження: функціональність систем дистанційного моніторингу якості питної води
4. Вихідні дані до роботи: методи, підходи та програмно-апаратні засоби здійснення дистанційного моніторингу якості питної води
5. Перелік програмного матеріалу: структура системи, копії екрану та код.
6. Зміст кваліфікаційної роботи: Вступ. Розділ 1. Огляд існуючих методів та засобів для комп'ютеризованого дистанційного моніторингу якості питної води. 1.1. Дистанційний моніторинг води. 1.2. Основні задачі дистанційного моніторингу води. 1.3. Джерела та види забруднення поверхневих вод. 1.4. Поняття комп'ютеризованого гідроекологічного моніторингу. Розділ 2. Проектування апаратних засобів комп'ютеризованої системи для контролю якості питної води. 2.1. Обґрунтування архітектурних рішень для реалізації задачі контролю якості питної води. 2.2. Структурна схема блоку моніторингу води. 2.3. Реалізація апаратної частини проектованої системи. Розділ 3. Програмне забезпечення комп'ютеризованої системи для моніторингу якості питної води. 3.1. Вибір засобів реалізації та середовища розробки проектованої системи. 3.2. Обґрунтування вибору хмарного середовища для збору та опрацювання результатів моніторингу якості питної води. 3.3. Програмне забезпечення для системи контролю якості питної води. Висновки. Список використаних джерел. Додатки. Додаток А

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури ДР	до 15.10.2023	15.10.2023
2	1. Вступ та I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	01.01.2024	01.04.2024
3	II розділ (аналіз експериментальної частини і методики досліджень)	03.02.2024	10.04.2024
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	10.03.2024	30.04.2024
5	Висновки	01.04.2024	06.05.2024
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	01.05.2024	25.05.2024
7	Отримання відгуку керівника	15.05.2024	30.05.2024
8	Отримання зовнішньої рецензії	17.05.2024	31.05.2024

8. Дата видачі завдання 20.10. 2023

9. Керівник кваліфікаційної роботи

10. Гарант

освітньо-професійної програми

11. Завдання до виконання отримав
здобувач вищої освіти



Горбовий А.



Омельчук А



Шмутьський С

АНОТАЦІЯ

Комп'ютеризована система дистанційного моніторингу якості питної води

Мета роботи: вдосконалення комп'ютеризованої системи для дистанційного моніторингу якості питної води.

Завдання, що мають бути вирішені при виконанні роботи:

- визначити цілі та завдання систем моніторингу якості питної води;
- проаналізувати систему моніторингу стану якості питної води;
- сформулювати критерії вибору системи для моніторингу води ;
- виконати аналіз комп'ютеризованої системи дистанційного моніторингу;
- вдосконалити комп'ютеризовану систему для дистанційного моніторингу якості питної води.

Об'єкт дослідження: контроль якості питної води. Засоби дистанційного моніторингу стану якості питної води.

Предмет дослідження: функціональність систем дистанційного моніторингу якості питної води.

Інформаційна база дослідження: методи, підходи та програмно-апаратні засоби здійснення дистанційного моніторингу якості питної води.

За результатами дослідження було сформульовано висновки про необхідність постійного моніторингу якості питної води, а також запропоновано програмні засоби, які розширяють функціональні можливості комп'ютеризованої системи контролю за якістю води.

Одержані результати можуть бути використані при формуванні комп'ютеризованих систем моніторингу якості питної води, в державних установах які спеціалізуються збереженням якості питної води.

Дипломна робота містить 42 сторінки, 26 рисунків, 1 таблиця і список використаних джерел з 40 найменувань.

Ключові слова: комп'ютеризована система дистанційного моніторингу, якості питної води, датчики, програмування.

Рік виконання дипломної роботи: 2023-2024

Рік захисту роботи: 2024.

ABSTRACT

Computerized system of remote monitoring of drinking water quality

Purpose of work: development of software for remote monitoring of drinking water quality.

The tasks that were solved during the performance of the work:

- the goals and objectives of drinking water quality monitoring systems are defined;
- types of drinking water quality monitoring software were analyzed;
- criteria for selecting software for water monitoring were formed;
- a comparative analysis of monitoring software was performed;
- the selection of tools for the computerized system of remote monitoring of the quality of drinking water was made;
- developed software for remote monitoring of drinking water quality.

Research object: drinking water quality control. Means of remote monitoring of drinking water quality.

The subject of research: the functionality of remote monitoring systems of drinking water quality.

Research information base: methods, approaches and hardware and software for remote monitoring of drinking water quality.

Based on the results of the research, conclusions were formulated about the need for constant monitoring of drinking water quality, and software tools were proposed that will expand the functional capabilities of the computerized water quality control system.

The obtained results can be used in the formation of computerized systems for monitoring the quality of drinking water in state institutions that specialize in maintaining the quality of drinking water.

The thesis contains 42 pages, 26 figures, 1 table and a list of used sources from 40 names.

Keywords: computerized remote monitoring system, drinking water quality, sensors, programming.

The year of completion of the thesis: 2023-2024 Year of job protection: 2024.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ КОМП'ЮТЕРИЗОВАНОГО ДИСТАНЦІЙНОГО МОНІТОРИНГУ ЯКОСТІ ПИТНОЇ ВОДИ.....	5
1.1. Дистанційний моніторинг води.....	6
1.2. Основні задачі дистанційного моніторингу води.....	7
1.3. Джерела та види забруднення поверхневих вод.....	8
1.4. Поняття комп'ютеризованого гідроекологічного моніторингу.....	8
РОЗДІЛ 2. ПРОЕКТУВАННЯ АПАРАТНИХ ЗАСОБІВ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ДЛЯ КОНТРОЛЮ ЯКОСТІ ПИТНОЇ ВОДИ.....	11
2.1. Обґрунтування архітектурних рішень для реалізації задачі контролю якості питної води.....	12
2.2. Структурна схема блоку моніторингу води.....	18
2.3. Реалізація апаратної частини проектованої системи.....	14
РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ПИТНОЇ ВОДИ.....	25
3.1. Вибір засобів реалізації та середовища розробки проектованої системи.....	25
3.2. Обґрунтування вибору хмарного середовища для збору та опрацювання результатів моніторингу якості питної води.....	29
3.3. Програмне забезпечення для системи контролю якості питної води.....	30
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТКИ.....	40

ВСТУП

Актуальність теми. Тривалість та якість життя людини напряму залежить від стану води, яку вона вживає. Різке збільшення урбанізації, ввід в експлуатацію нових промислових підприємств, які забруднюють землю та водойми – ці фактори впливають на якість питної води та погіршують екологічний стан довкілля, саме для швидкого реагування та виявлення місця забруднення водних ресурсів є актуальною розробка комп'ютеризованої системи дистанційного моніторингу якості питної води.

Якість питної води є необхідним фактором для здорового людей. Широке застосування засобів дистанційного моніторингу є ефективним інструментом для виявлення джерел забруднення питної води. Воно полягає у визначенні рівня концентрації шкідливих речовин у воді в режимі реального часу.

Існуючі в Україні технічні засоби не забезпечують в повній мірі потреб у моніторингу якості води з точки зору швидкості отримання даних та можливостей щодо їх аналізу та опрацювання. Тому розроблення комп'ютеризованої системи дистанційного моніторингу якості питної води є актуальним завданням, яке визначило напрям дослідження кваліфікаційної роботи.

Мета і завдання дослідження. Метою роботи є вдосконалення комп'ютеризованої системи для підвищення ефективності контролю якості питної води завдяки застосуванню методів дистанційного моніторингу.

Завдання які потребують потребують вирішення для досягнення мети роботи:

- визначити цілі та завдання систем моніторингу якості питної води;
- проаналізувати систему моніторингу стану якості питної води;
- сформулювати критерії вибору системи для моніторингу води ;
- виконати аналіз комп'ютеризованої системи дистанційного моніторингу;
- вдосконалити комп'ютеризовану систему для дистанційного моніторингу якості питної води.

Об'єкт для дослідження – комп'ютеризована система дистанційного моніторингу якості питної води.

Предмет для дослідження – програмно-апаратні методи та засоби побудови комп'ютерних систем для дистанційного моніторингу якості води.

Методи для дослідження – системний аналіз, синтез, порівняння та узагальнення використовуються у кваліфікаційній роботі задля виконання задач

Наукова новизна отриманих результатів:

1. Метод роботи з даними вдосконалено в комп'ютеризованій системі дистанційного контролю якості питної води та забезпечено можливість в дистанційному режимі визначати рівень забрудненості води, підвищується швидкість виявлення джерел викиду шкідливих речовин та запобігти їх поширенню.

2. Завдяки застосуванню створеного програмного забезпечення, підвищується швидкість визначення рівня забрудненості, що дає змогу покращити екологічний стан довколишнього середовища.

Практичне значення одержаних результатів кваліфікаційної роботи полягає у тому, що запропоновані та реалізовані програмно-апаратні засоби дозволяють в режимі реального часу контролювати якість питної води і надавати актуальну інформацію профільним організаціям, що слугитиме ефективним інструментом для покращення екологічного стану довкілля.

Структура роботи.

Кваліфікаційна робота містить 42 сторінки, 26 рисунків, 1 таблиця і список використаних джерел з 40 найменувань.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ КОМП'ЮТЕРИЗОВАНОГО ДИСТАНЦІЙНОГО МОНІТОРИНГУ ЯКОСТІ ПИТНОЇ ВОДИ

1.1. Дистанційний моніторинг води

Дистанційний моніторинг води - це аналіз і спостереження параметрів, за допомогою комп'ютеризованих систем, що дають змогу на відстані контролювати за якістю води. Даний процес дозволяє оцінювати і контролювати якість водних ресурсів для захисту навколишнього середовища, захисту здоров'я людини і підтримки постійного використання систем водопостачання.

Основними складовими дистанційного спостереження за якістю питної води є збір проби води та вимірювання параметрів, таких як рН, розчинені речовини, шкідливі речовини та бактерії через датчики, які передають дані у комп'ютеризовану систему в режимі реального часу. Даний процес включає в себе такі вимірювання як польові та лабораторні, використання спеціалізованих приладів та пристроїв, використання аналітичних методів. [1]

Аспекти потребуючі у комп'ютеризованій системі дистанційного моніторингу якості питної води.

Охорона здоров'я:

Система дистанційного моніторингу якості питної води може допомогти вчасно визначити чи наявні шкідливі речовини, віруси та бактерії у воді. Все це запобігає розповсюдженню всіляких водних хвороб, а саме, гепатит С, дизентерія та інші, і допомагає підтримувати здорове населення людей.

Охорона навколишнього середовища:

Дистанційний моніторинг якості питної води допомагає визначити забруднення джерел води, спричинене промисловими відходами, господарською діяльністю та іншими джерелами забруднення. Все це дозволить вам вжити діючих заходів від зараження.

1.2. Основні задачі дистанційного моніторингу води

Дистанційний моніторинг водних ресурсів має багато завдань, які спрямованих на аналіз інформації стану якості питної води. [1]

Визначення якості питної води: Головним завданням дистанційного моніторингу - є визначення біологічних та хімічних властивостей води. В даний параметр входить виміри таких параметрів, а саме, рівень розчинених речовин, рН, температура, концентрація кисню, пестицидів та інших шкідливих речовин.[2]

Виявлення забруднюючих речовин: дистанційний моніторинг якості питної води допоможе вчасно виявити саме наявність та місце забруднення води. Таким місцем може опинитись промисловий стік шкідливих речовин, господарські відходи, стічні води та інші джерела забруднення. Знаходження забрудненої ділянки дозволяє швидко зреагувати та вдіяти заходи щодо зменшення впливу на навколишнє середовище та забезпечити відновлення екосистеми.

Екологічна оцінка: дистанційний моніторинг якості питної води дозволить дати екологічну оцінку стану води та екосистеми. Все це включає до себе визначення біорізноманіття, оцінку екосистеми, вчасне виявлення змін та тенденцій у навколишньому середовищі.

1.3. Джерела та види забруднення поверхневих вод

Води бувають забруднені різними способами та різними речовинами. Основні джерела забруднення водних ресурсів саме є :

Індустріальні відходи: це відходи, які можуть зливатися з всіляких промислових підприємств та містять дуже небезпечні властивості, відходи промислового виробництва та хімічні речовини.

Господарське забруднення: це забруднення від сільсько-господарської діяльності вирощання добрив та іншої хімії, що може потрапляти у воду через зливи, рознос ґрунту та через неправильний злив стічної води.

Комунальні міські стічні води: Скидання міського стоку та інших населених пунктів, може містити в собі бактерії, віруси, хімічні речовини та інші забруднюючі речовини.

Незаконне скидання: Незаконні зливи це не правильний спосіб утилізації отруйних відходів промисловості та інших брудних видів сміття без відповідних методів очищення та переробки.

Типами забруднення води може змінюватися залежно від забруднення та його виду. Основними типами забруднення води є:

Біологічне забруднення: таке забруднення може бути визване через забруднення бактеріями, вірусами та іншими інородними організмами, що може бути присутнє в стічних водах.

Хімічне забруднення: сюди входять різні хімічні речовини, а саме, токсичні речовини, пестициди, та інші хімікати. Дані речовини мають дуже шкідливий вплив на екосистему та здоров'я людини.

Різні сполуки: Використання мінеральних добрив у відходах від господарства призведе до витіку фосфору та азоту у води поверхні. Це призведе до евтрофікації, резервуар води стає перенасиченим поживними речовинами, що може сприяти надмірному росту водоростей та інших тіл, це негативно вплине на екосистему води.

Такі види забруднення сприяють появі серйозних проблем, а саме, якість питної води, вимирання риб та серйозні зміни в екосистемі і негативний вплив на здоров'я людини. Контроль та дистанційний моніторинг забруднення води є важливою задачею для зберігання екосистеми та сталого використання водних ресурсів. Саме для цього використовуються різні методи, включаючи регулювання стоку забруднених речовин, регулювання системи для очищення питної води. Крім цього, важливо систематично виконувати гідроекологічний дистанційний моніторинг для оцінки якості питної води, для вчасного виявлення змін та розробки плану для ефективного управління водними ресурсами.

1.4. Поняття гідро-екологічного моніторингу

Дистанційний моніторинг - це система збору, аналізу та оцінки даних про якість питної води та її стан саме з метою визначення екологічного стану, виявлення забруднення, моніторингу за ефективністю заходів та розробка наукових рекомендацій щодо збереження та відновлення системи водних ресурсів. [3]

Дистанційне гідро-екологічне спостереження включає в себе аналіз хімічних, біологічних та гідро-екологічних даних. Зазвичай для моніторингу використовуються такі методи дослідження, а саме, збір проби води, вимірювання

її рівня, температури, концентрації розчинених речовин, вмісту кисню, виявлення шкідливих речовин та інше.

Метою екогідрологічного дистанційного моніторингу є збір інформації про стан водної екосистеми, виявлення зміни у якості питної води та їхнього впливу на екосистему. Ці фактори дозволяють прийняти рішення щодо збереження та віднову водних об'єктів, а також моніторингу та дотримання норм якості питної води та вживання заходів задля її поліпшення.

Гідро-екологічний дистанційний моніторинг є важливою частиною екологічного менеджменту водних ресурсів і допомагає забезпеченню сталого використання та збереження водних систем для майбутнього покоління.

РОЗДІЛ 2. ПРОЕКТУВАННЯ АПАРАТНИХ ЗАСОБІВ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ДЛЯ КОНТРОЛЮ ЯКОСТІ ПИТНОЇ ВОДИ

2.1. Обґрунтування комп'ютерних рішень для реалізації задачі контролю якості питної води.

Щоб реалізувати мету даної кваліфікаційної роботи розроблено схему для розуміння взаємодії передачі даних до комп'ютеризованої системи моніторингу якості питної води, (рис. 2.1).

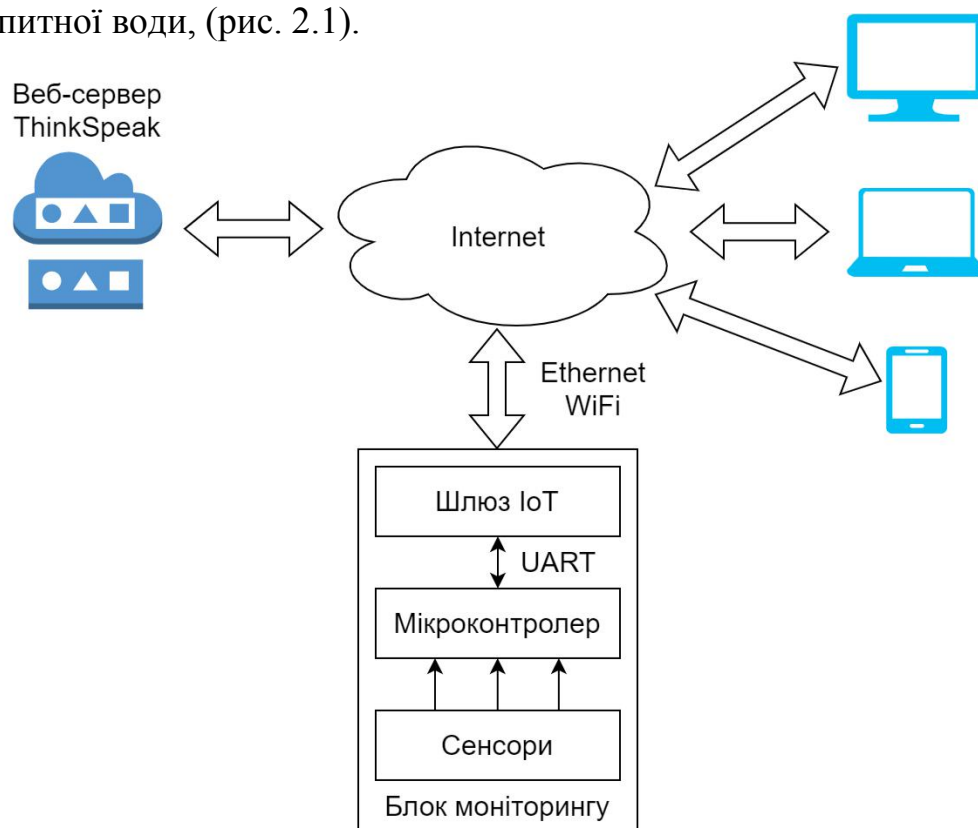


Рис. 2.1. Структурна схема інформаційно-вимірювальної системи для контролю якості питної води

На схемі зображеній на (рис.2.1). структуру передачі даних з сенсору, за допомогою датчика до мікроконтролера, той в свою чергу передає дані до шлюзу IoT для зберігання та через мережу інтернет дані за допомогою розробленої комп'ютеризованої системи потрапляють на веб-сервер та стають доступними для перегляду.

Мікроконтролер у даній системі прикладного рівня складається виконує функцію приймання результату датчиків від модуля і відразу надсилає їх на хмарний сервер IoT.

Дана система складається з мікрокомп'ютера, що відповідає моніторингу якості питної води, та містить модуль Adruino Uno та датчика підключеного для виміру якості води .

Сервер, застосований на платформі Thingspeak для інтернет речей, призначених задля зберігання даних вимірювань, обробки даних та його аналізу, з використанням можливостей MATLAB, а також для відображення їх у графічній формі в зручному для користувача форматі.

2.2. Структурна схема блоку моніторингу води

Структурна схема блоку дистанційного моніторингу якості питної води виконана на основі аналізу завдання кваліфікаційної роботи та огляду її аналогів

Основні компоненти, які входять до складу блоку:

- мікрокомп'ютер;
- мікроконтролерний модуль;
- датчики.

Основні дії блоку для моніторингу якості питної води полягає в отриманні даних від датчиків мікроконтролеру та швидка передача до Raspberry Pi, який вчасно надсилає дані через IP-мережу (Wi-Fi або Ethernet) у хмарну платформу IoT ThingSpeak для дистанційного моніторингу.

Модуль Raspberry Pi використовується для шлюзу між IoT для передавання даних на віддалений сервер, з використанням IP-протоколу. Модуль Arduino Uno в цьому проекті вимірює забруднення водних ресурсів через цифрові датчики. Цей модуль відповідає управлінню режимами системи, що стосуються обміну даними. Raspberry Pi – цей модуль використовується для отримання інформації

від модулю Arduino Uno, керування процесом надсилання даних у віддалений сервер.

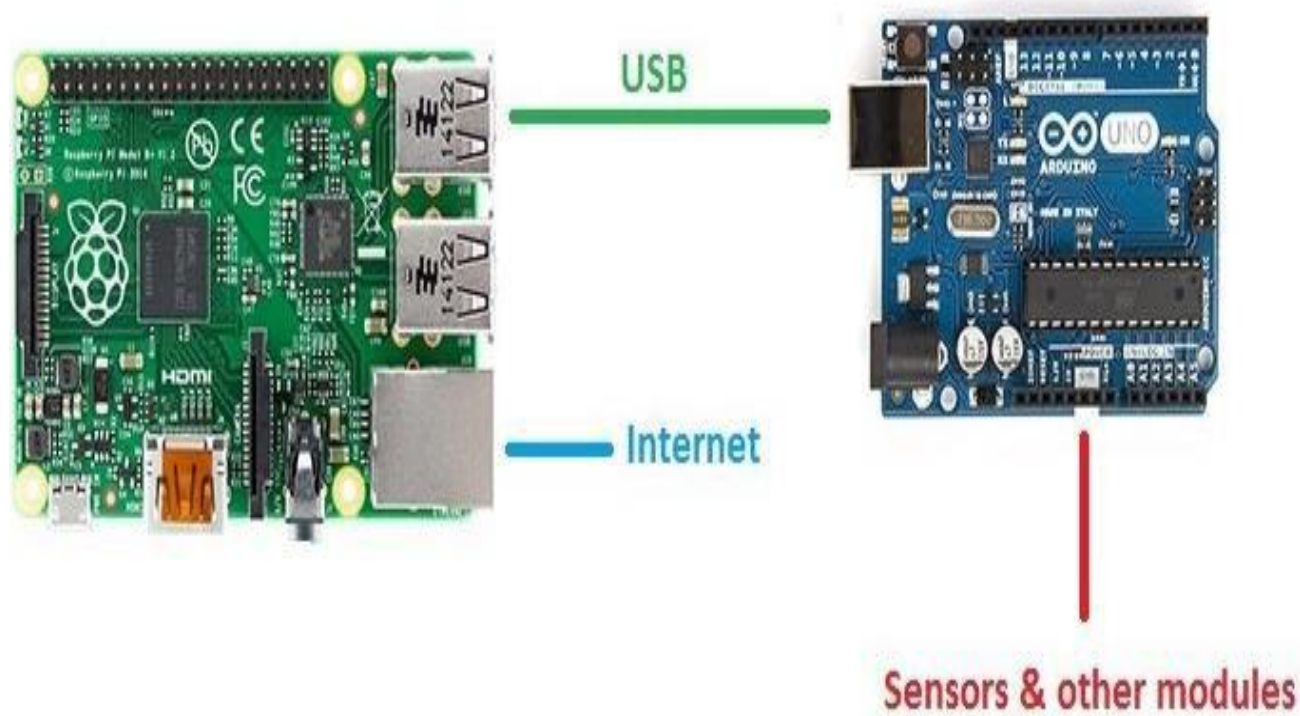


Рис. 2.2. Дії модулів між собою Raspberry Pi та Arduino Uno

2.3. Реалізація апаратної частини проектованої системи

Опис модуля Raspberry Pi.

Raspberry Pi (рис. 2.3) – це модуль, малого розміру, що працює на операційній системі Linux. Даний модуль забезпечує рішення для дистанційного моніторингу контрольованих. Raspberry Pi 3 має чотирьох-ядерний шестидесяти чотирьох розрядний мікропроцесор Cortex A53, з тактовою частотою 1,4 гігагерц, 1 гігабайт SDRAM з вбудованим Bluetooth та Wi-Fi.

В якості ПЗУ для RPi використовується карта пам'яті обсягом 16 гігабайт. Raspberry Pi модуль може також бути використаним зі звичайними периферійними пристроями ПК, такими як клавіатура, миша, до даного модуля можна також підключити монітор за допомогою HDMI роз'єму. Для включення мікрокомп'ютера потрібно подати живлення за допомогою microUSB роз'єму.



Рис. 2.3. Вигляд плати мікрокомп'ютера RPi 3

Мікрокомп'ютер дає можливість передачі даних із застосуванням технологій Ethernet, що ставились перед системою. Роз'єм мікрокомп'ютеру Raspberry Pi має роз'єм, що складається із сорока цифрових виводів загального призначення (Додаток А).

Крім того 27 (BCM 0) і 28 (BCM 1) піни використовуються задля налаштування модулю Raspberry Pi для роботи з поверхневими пристроями, вони являються звичайними платами, для використання даних пінів. Вони є повноцінними GPIO-пінами. Фактично виходить, що GPIO-пінів не 40, а 28.

Під час розробки програмного забезпечення, потрібно точно вказати режим нумерації, який буде користаний, щоб попередньо запобігти виникненню проблем.

Варіанти нумерації пінів модуля Raspberry Pi: Board (фізична нумерація по порядку) і BCM (нумерація з чіпу).

Контакти GPIO RPi є актуальними при застосуванні RPi для систем IoT. На апаратному рівні підтримуються інтерфейси I²C, SPI, UART і 1-wire.

Кожен з використаних пінів має можливість бути встановленим у цифровий вихід OUTPUT, та в цифрового вхід INPUT. Кількість максимального вихідного струму кожного з пінів не має перевищувати 16 мА. Сумарний вихідний струм всіх пінів не повинен перевищувати 50 мА. Значно більший струм здатен

видавати 5-вольтовий пін, який використовується у живленні модулю Raspberry Pi 3 та інших пристроїв (клавіатури, миші) – до 500 мА.

За замовчуванням всі піни (крім BCM14 і BCM15) знаходяться в режимі INPUT, а в цей час піни BCM0-BCM8 і BCM15 підтягнуті резисторами (pullup). Мультиметр може показати високий рівень напруги на цих пінах, саме через цю причину. Решта пінів стягуються до нуля.

Кожен з 28 пінів забезпечений двома резисторами: підтягуючим (pullup) і стягуючим (pulldown), в режимі INPUT може бути стягнутий до нуля. Номінал опору підтягуючого / стягуючого резистора для пінів BCM2 і BCM3 – 1,8 КОм. Кожен з 28 пінів в режимі INPUT може генерувати переривання, а також в асинхронному режимі по фронту і по спаду.

RPi 3 не має аналогових входів / виходів. Для реалізації подібного функціоналу потрібно використовувати зовнішні пристрої з АЦП / ЦАП. Саме тому, для проектування системи для контролю якості питної води було прийнято рішення в використати модуль Arduino Uno.

Опис модуля Arduino Uno.

Модуль Arduino Uno є одним з найважливіших апаратних компонентів блоку моніторингу якості води (Додаток В). Arduino Uno – це одна з найпопулярніших плат для створення прототипів комп'ютеризованих систем.

Головною перевагою данного модуля є простий процес зміни програмного коду завдяки наявності вбудованого програматора з прямим підключенням до ПК без необхідності використання додаткових компонентів. Платформа Arduino Uno побудована на базі мікроконтролера ATmega (рис. 2.6).

Вона містить шість аналогових входів, чотирнадцять цифрових входів / виходів, роз'єми USB та ICSP і кнопку перезавантаження. Напруга живлення для плати Arduino Uno може подаватися від зовнішнього блока живлення або через USB роз'єм від системного блока ПК. Спосіб подачі електроенергії до плати Arduino Uno вибирається виходячи з особливостей проекту, в якому планується її використання.

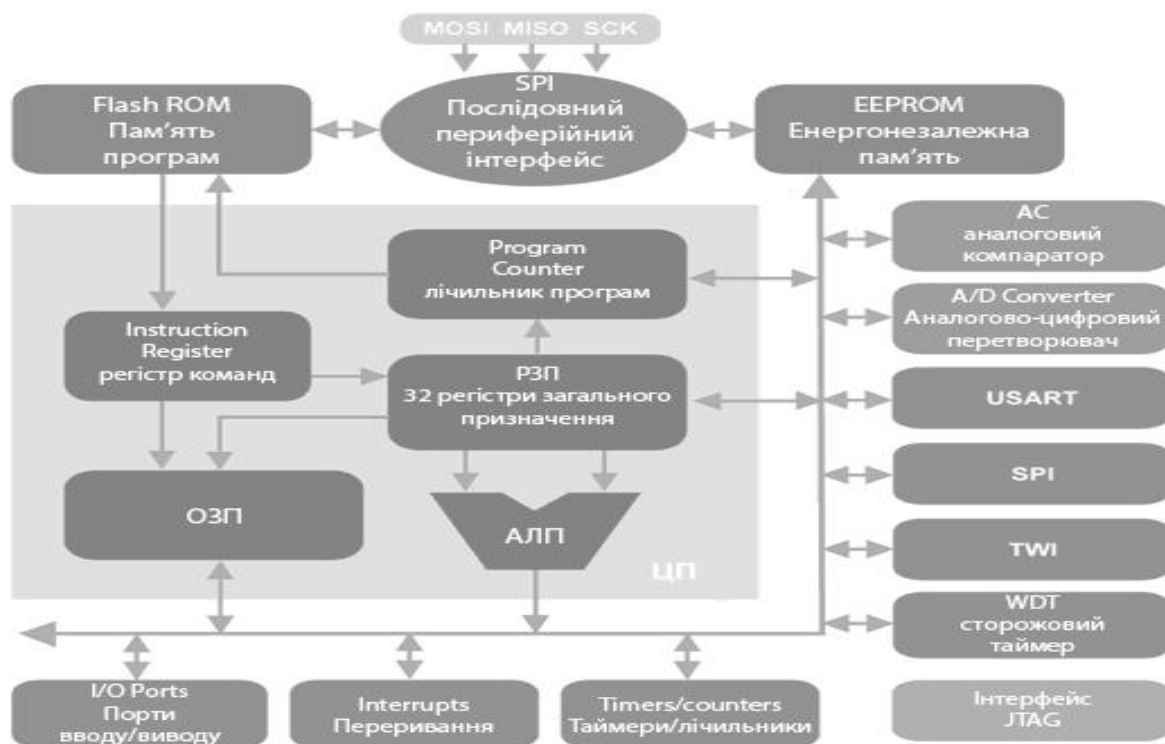


Рис. 2.4. Функціональна схема мікроконтролера

Модуль Sensor W-25011 датчика якості води призначений для виявлення дрібних часточок з розміром більшим за 0,8 мікрметра. Модуль характеризується низьким рівнем споживання енергії, який можна знизити шляхом вимкнення підсвічування пристрою зчитувача (рис. 2.8).

Широкий діапазон вхідної напруги живлення забезпечується вбудованим стабілізатором.

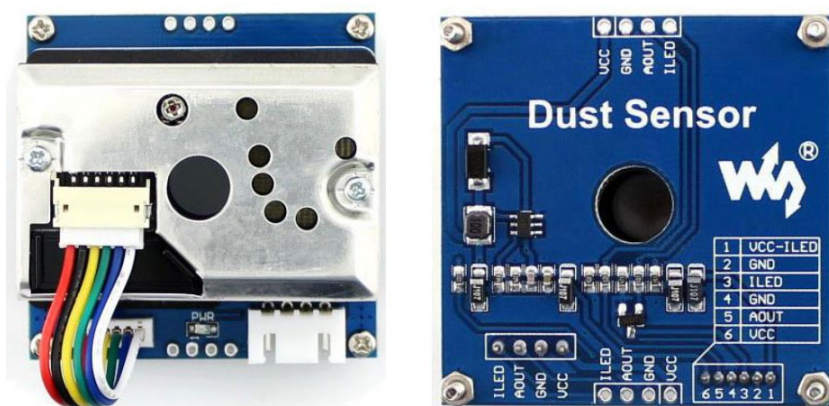


Рис. 2.5. Зовнішній вигляд модуля Sensor W-25011 датчика

Датчик використовується в системах контролю рівня забруднення, кондиціонування та очищення води. Технічні характеристики датчика Sensor W-25011 F наведені в табл. 2.1.

Таблиця 2.1.

Технічні характеристики датчика Sensor W-25011

Назва характеристики	Значення
Діапазон вимірювання	500 мікрограм/м ³
Чутливість	500 мВ / (100 мікрограм/м ³)
Максимальний робочий струм	0,02 А
Напруга живлення	2500 мВ - 5500 мВ
Робоча температура	від -10 °С до +65 °С

Електрична схема модуля датчика Sensor W-25011 показана в Додатку С. Модуль датчиків WQC1090 і CW911 якості води.

Модуль об'єднує два датчики, які розміщені на одній платі (рис. 2.10). Результати вимірювання обох датчиків можуть бути отримані окремо завдяки використанню I²C інтерфейсу. WQC1090 – це мініатюрний цифровий датчик. CW911 – це високоточний, мініатюрний, високоефективний цифровий датчик вологості і температури. Датчик оксиду металу (МОКС) призначений для вимірювання концентрації летючих органічних речовин eTVOCs та вмісту CO₂. В середині модуля вбудований мікроконтролер, який на основі отриманих даних про рівень вологості і температури передавача WQC1090 здійснює корекцію результатів вимірювання вмісту газу, які вимірюються датчиком CW911.

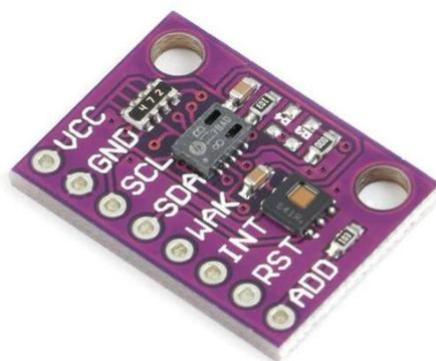


Рис. 2.6. Зовнішній вигляд модуля датчиків WQC1090 і CW911 якості води

Модуль датчика якості води MQ-135.

Модуль датчика якості питної води MQ-135 застосовується в даній системі для визначення рівня концентрації шкідливих речовин у воді (рис. 2.12). Розроблений на базі датчика MQ135. На платі модуля розміщений компаратор, який призначений для порівняння рівня концентрації шкідливих речовин з пороговим значенням, поріг спрацювання якого налаштовується потенціометром.



Рис. 2.7. Зовнішній вигляд модуля датчика якості води MQ-135

Модуль MQ-135 за допомогою компаратора визначає рівень шкідливих речовин у воді та порівнює їх з нормою та передає інформацію до комп'ютеризованої системи.

Характеристики: напруга живлення датчика становить 5 В;
на виході датчик видає аналогову напругу 0 – 5 В пропорційно до рівня рівні вихідних сигналів передавача відповідають TTL логіці;
Принципова схема роботи модуля для дистанційного моніторингу якості води MQ-135 зображена на (рис. 2.8).

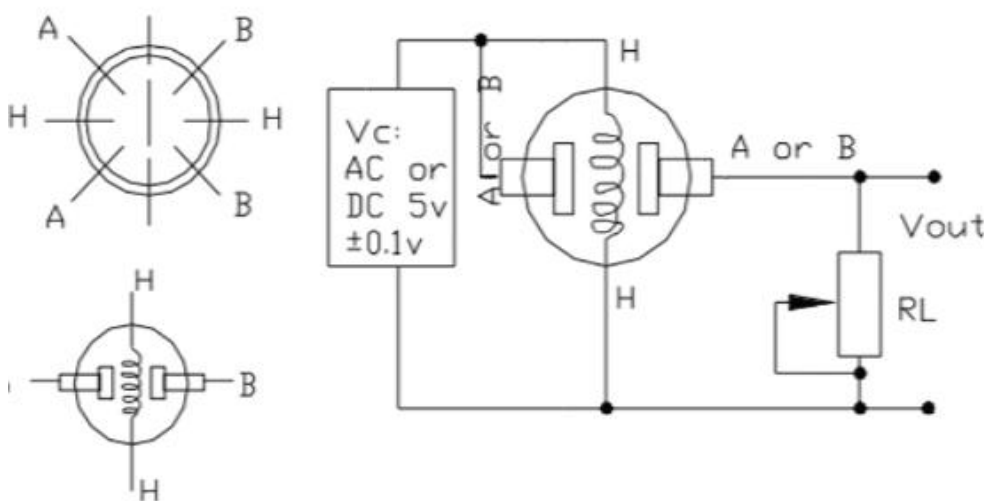


Рис. 2.8. Схема електрична принципова модуля датчика якості води MQ-135

Опис електричної схеми.

Схема електрична принципова блока моніторингу якості води (Додаток С).

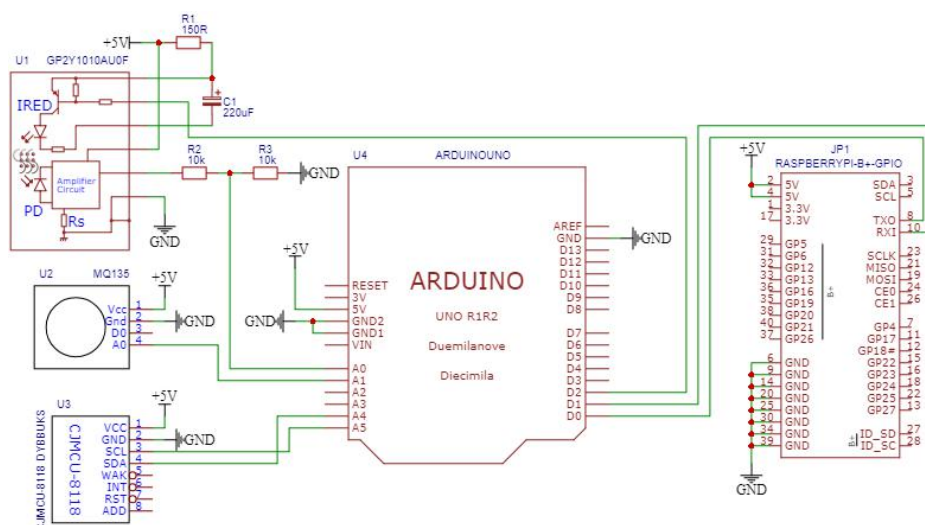


Рис. 2.9. Схема електрична принципова блока для моніторингу якості води

В процесі розробки проектованої системи виникла задача об'єднання Arduino Raspberry Pi в одному пристрої, щоб скористатися перевагами відразу обох платформ. В цьому проекті є датчики, які можна підключити до Arduino (використовуючи його можливості роботи в режимі реального часу), а дані від датчиків обробляти і передавати по мережі інтернет з використанням Raspberry Pi (використовуючи його обчислювальні та мережеві можливості). Таким чином, будуть відокремлені обчислювальні та комунікаційні задачі (виконуються Raspberry Pi) і задачі моніторингу (виконуються Arduino). В цій схемі модуль датчиків якості води U3 під'єднується до плати Arduino Uno по шині I²C. А передавачі U1 і U2 під'єднані до входів АЦП A0 та A1 відповідно.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ПИТНОЇ ВОДИ

3.1. Вибір засобів реалізації та середовища розробки проекрованої системи
Обґрунтування вибору операційної системи для Raspberry Pi.

На даний момент для Raspberry Pi існує шість офіційних операційних систем близько 30 неофіційних, включаючи в тому числі і Windows 10 IoT Core та Android Raspberry Pi. Проте в більшості випадків Raspberry Pi (RPi) працює на ОС, які базуються на ядрі Linux, таких як: Raspberry Pi OS (раніше називалась Raspbian), Arch Linux ARM, Pidora (модифікація Fedora), Kali Linux.

ОС Raspberry Pi OS – це найстабільніша модифікація найпоширенішої операційної системи Linux Debian / GNU. Вона є офіційною версією операційної системи для цих одноплатних мікрокомп'ютерів [18]. Саме тому, в проектованій системі для мікрокомп'ютера RPi було прийняте рішення встановити ОС Raspberry Pi OS, в якій є можливість використовувати легкий графічний інтерфейс (LXDE), а також стандартний менеджер пакетів для встановлення програм через Інтернет (рис. 3.1).

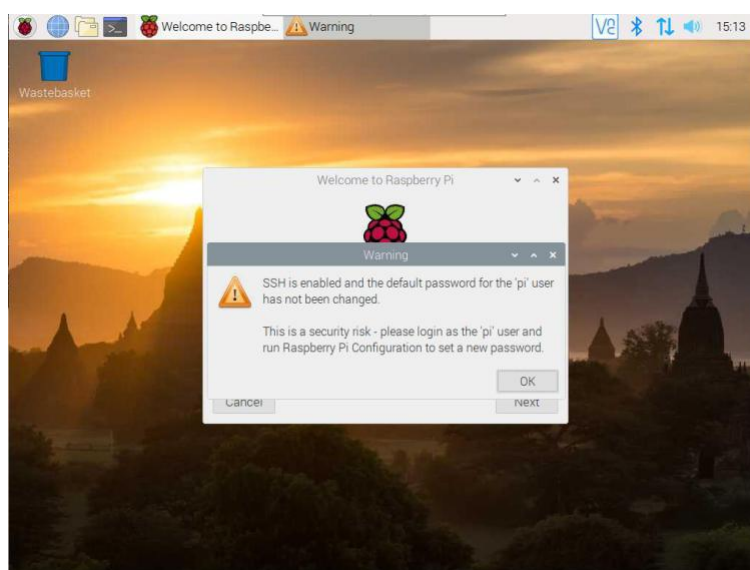


Рис. 3.1. Вигляд робочого столу операційної системи Raspberry Pi OS

Для встановлення та налаштування Raspberry Pi OS використана карта пам'яті типу microSD 10 класу швидкості і картрідер. Карту пам'яті було відформатовано як FAT32, використовуючи безкоштовну утиліту SD Card Formatter. Файли ОС Raspberry Pi OS були завантажені з офіційного сайту Raspberry Pi Foundation [18], та записані на microSD-карту об'ємом 16 Гб за допомогою утиліти Win32DiskImager з графічним інтерфейсом.

Для роботи з RPi був використаний «headless» підхід, який передбачає що замість підключення до мікрокомп'ютера пристроїв введення/виведення (монітора, миші та клавіатури) взаємодія з RPi відбувається з використанням ПК, який знаходиться в спільній ним локальній комп'ютерній мережі [19]. Це зроблено з застосуванням мережевого протоколу SSH для зручного захищеного доступу [20] до RPi. Для встановлення з'єднання SSH використовувалось ПЗ PuTTY [21], яке дозволяє працювати на віддаленому RPi та інших комп'ютерах з ОС Linux з застосуванням ПК на ОС Windows (рис. 3.2).

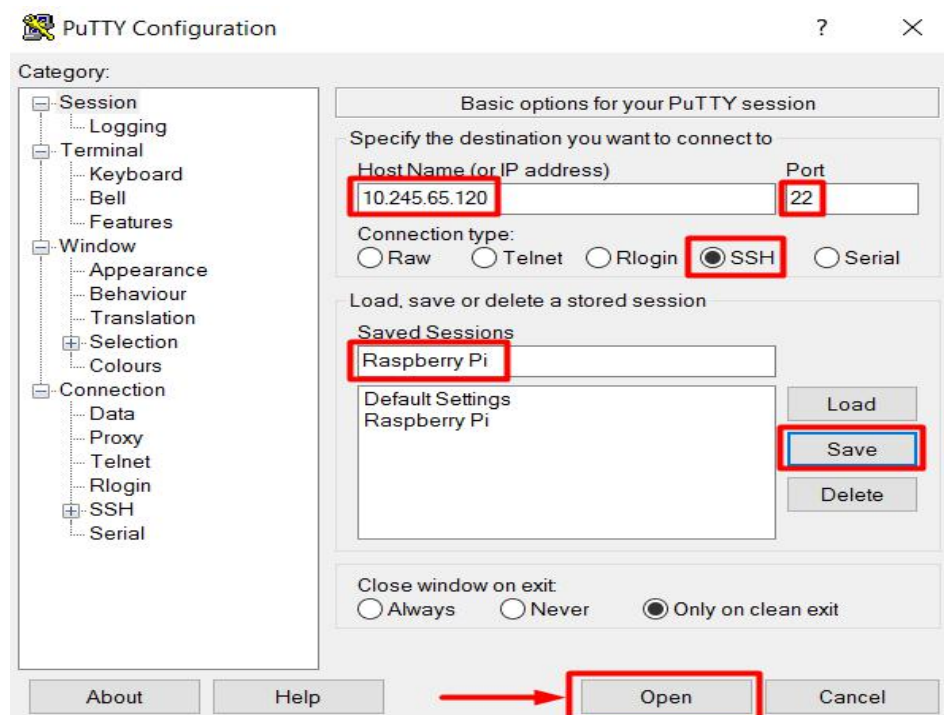


Рис. 3.2. Налаштування з'єднання SSH з Raspberry Pi OS
Вибір мови програмування та середовища розробки.

Програмувати Raspberry Pi можна на великій кількості різних мов – Java (Pi4J), Pascal, Perl, Ruby, C#, C, C ++, Basic та ін. Для реалізації задач кваліфікаційної роботи було прийняте рішення використовувати мову Python 3 версії разом із середовищем розробки Thonny Python IDE, яке встановлене в Raspberry Pi OS (рис. 3.3).

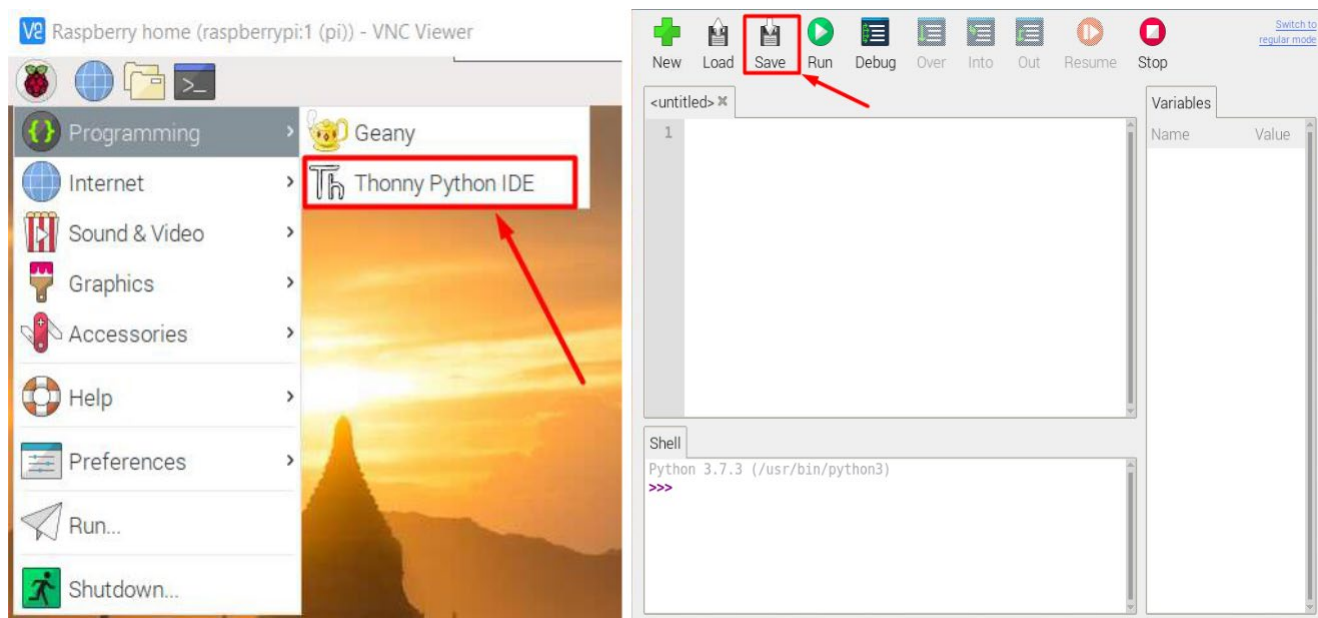


Рис. 3.3. Вигляд програмного середовища в Raspberry Pi OS для написання коду на мові Python

Скрипти на Python можна також писати в текстовому редакторі командного рядка ОС. В процесі написання програм були підключені додаткові бібліотеки та класи:

бібліотека «RPi.GPIO» для роботи з цифровими виводами Raspberry Pi;

бібліотека «pySerial» для UART інтерфейсу;

клас «time» для організації часових затримок.

Вся взаємодія програмного і апаратного забезпечення, тобто створення програмного забезпечення, з подальшим завантаженням в мікроконтролер, відбувалася з використанням операційної системи Windows 10.

Для написання програмного коду для модуля Arduino було обрано середовище Arduino IDE найновішої версії [22]. В цьому середовищі програмування застосовується компілятор AVR-GCC, тому програмування

мікроконтролера ATmega328P здійснювалось на мові програмування C++ (рис.3.4).



Рис. 3.4. Зовнішній вигляд робочої області середовища розробки Arduino

Вибір середовища Arduino IDE для розробки проектованої системи обумовлений багатьма його перевагами, зокрема:

- відсутність необхідності у великому об'ємі пам'яті;
- невимогливість до ресурсів комп'ютера; простота та зручність у користуванні;
- легкий процес підключення до плати;
- повна сумісність з операційними системами Windows;
- наявність вибору кількох мов програмування;
- достатня кількість функцій та інструментів, необхідних для роботи.

Програма, яка написана в Arduino IDE називається скетчем. Перед першою компіляцією, програма повинна бути збережена в окремому файлі на комп'ютері. Кожен скетч базується на шаблоні, який складається з 2 блоків – «setup» та «loop».

Перший блок являє собою функцію, яка виконується лише раз при запуску програми і дозволяє налаштувати піни, здійснити ініціалізацію певних змінних тощо. А другий блок є основною частиною програми. Код в ньому виконується циклічно. В цій частині описуються всі основні функції. Цикл буде виконуватись періодично до тих пір, поки пристрій під'єднаний до джерела живлення.

3.2. Обґрунтування вибору хмарного середовища для збору та опрацювання результатів моніторингу якості питної води

ThingSpeak – це хмарна платформа для інтернету речей, яка дає змогу збирати, відображати і виконувати аналіз поточкових даних. В ThingSpeak є можливість отримувати дані з різних обчислювальних пристроїв, таких як ESP, Arduino, Raspberry Pi та інших подібних платформ. Крім того є можливість отримувати дані з ПК. Також платформа ThingSpeak забезпечує можливість відображення даних в режимі реальному часу [23].

Платформа ThingSpeak надає можливість інформаційно-вимірювальним пристроям, які отримують дані від передавачів, зберігати їх у хмарі – у публічних або закритих каналах. За замовчуванням канали ThingSpeak є закритими від інших користувачів, але за необхідності їх можна зробити доступними для всіх.

Відразу після надходження даних до каналу ThingSpeak, їх можна візуалізувати, перетворювати та аналізувати.

Ключові особливості даної хмарної платформи ThingSpeak [23]:

- налаштування пристроїв і надсилання даних в ThingSpeak через MQTT або REST API;

- збір даних за запитом з пристроїв та інших джерел;

- відображення поточних і історичних даних в реальному часі;
- попередній аналіз і опрацювання зібраних даних із застосуванням вбудованих функцій MATLAB;
- запуск аналітики при настанні події або в автоматичному режимі по таймеру;
- взаємодія з соцмережею Twitter і сервісом Twilio для відправлення повідомлень.

Зберігання інформації у хмарі забезпечує швидкий до ступ до неї. Застосування онлайн-інструментів аналітики дає змогу відображати дані в графічній формі та здійснювати їх дослідження, будувати взаємозв'язки, шаблони та тренди. З існуючих даних можна створювати нові, а також будувати різноманітні діаграми та графіки.

Система аналітики MATLAB, яка вбудована в ядро ThingSpeak, дає змогу писати і запускати MATLAB-код для виконання попереднього опрацювання, аналізу і візуалізації даних. ThingSpeak дозволяє проектувати IoT-системи без налаштування серверів і розробки веб-додатків. ThingSpeak надає доступ до функцій MATLAB, які дозволяють:

- здійснювати обчислення за наперед складеним розкладом; обробляти, об'єднувати і перетворювати дані;
- об'єднувати дані, отримані з декількох каналів, для проведення складнішого аналізу;
- візуально досліджувати взаємозв'язки між інформаційними даними, використовуючи вбудовані графіки.

Основою платформи ThingSpeak є канали, в які користувачі передають дані для зберігання та візуалізації. Кожен канал містить (рис. 3.5):

- два поля для імені та опису каналу;
- вісім полів для інформації будь-якого типу;
- три поля для географічних координат (висота, довгота, широта);
- поле для посилання на сайт, на якому буде використовуватися інформація, яка зберігається в каналі;

-посилання на YouTube канал.

ThingSpeak™ Channels Apps Devices Support

New Channel

Name

Description

Field 1 ☒

Field 2 ☒

Field 3 ☒

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

Metadata

Tags
(Tags are comma separated)

Link to External Site

Link to GitHub

Elevation

Show Channel Location ☐

Latitude

Longitude

Show Video ☐

☒ YouTube ☐ Vimeo

Video URL

Рис. 3.5. Діалогове вікно ThingSpeak для створення нового каналу

Сервіс ThingSpeak дає можливість взаємодії з різними пристроями. Наприклад, можна запрограмувати реакцію на отримані дані одразу після їх надходження до каналу. Крім того, можна створювати черги з команд для їх виконання на інших пристроях.

Зареєструвавши канал у ThingSpeak, до нього відразу ж можна надсилати інформацію, обробляти її та отримувати до неї доступ. Після реєстрації сервіс згенерує для каналу унікальний ідентифікатор Channel ID. Канали можуть працювати з інформацією у форматах CSV, XML та JSON.

Поля каналу можуть містити як цілі, так і дробові числові значення – це власне ті цифрові дані, які будуть надсилатися на сервер. Кожен квант даних має два виміри: значення і час. Зі значенням все більш-менш зрозуміло, а ось поле

часу встановлюється сервером автоматично. На нього не можливо вплинути. Більше того, час кожної порції даних визначається у форматі UTC+0, а вже при виведенні він форматується відповідно до часової зони користувача.

Платформа IoT ThingSpeak використовує методи API REST, такі як POST, GET, PUT та DELETE, для створення та видалення каналів, зчитування та запису даних каналів та очищення інформації у каналі. REST – це архітектурний стиль, розроблений як модель відповіді на запит, яка спілкується через HTTP. Клієнт або веб-браузер надсилає запит на сервер, який дає відповідь з даними у відповідному форматі. Веб-браузери використовують цей інтерфейс для отримання веб-сторінок або для надсилання даних на віддалені сервери. Також існує можливість використання методу MQTT Publish для оновлення каналу, або для того, щоб отримувати повідомлення, якщо канал оновлюється. API REST можна використовувати для створення та оновлення каналів та графіків ThingSpeak. API MQTT застосовується лише для оновлення каналів ThingSpeak.

Перевірити роботу створеного каналу можна, записавши інформацію за допомогою GET-запиту, використовуючи будь-який сучасний інтернет-браузер. Для цього потрібно використовувати ключ API Key – довгий буквено-цифровий код, який знаходиться на вкладці API Keys каналу. Ключі API Key для читання та запису є різними (рис. 3.6).

Channel ID: 1604471
Author: mwa000018132791
Access: Private

Application for air quality remote control based on Raspberry Pi. Created by Lariomyk Roman

Private View Public View Channel Settings Sharing **API Keys** Data Import / Export

Write API Key

Key: EI3MU9PVGK0HP4TW

Generate New Write API Key

Read API Keys

Key: H7TT43TZ5RNVQ1G

Note:

Save Note Delete API Key

Add New Read API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=EI3MU9PVGK0HP4TW&field=
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1604471/feeds.json?api_key=
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/1604471/fields/1.json?api_key=
```

Рис. 3.6. Вікно вкладки API Keys платформи ThingSpeak

Наприклад, для очищення каналу за допомогою HTTP DELETE необхідно створити файл в форматі JSON з ідентифікатором користувача та передати його на дану адресу:

```
https://api.thingspeak.com/channels/<channel_id>/feeds.json
```

де `<channel_id>` – ідентифікатор каналу, `feeds.json` – назва JSON файлу.

На кожен запит сервер дасть розгорнуту (JSON) відповідь, де будуть вказані Channel ID, значення всіх полів, дата та час. Якщо файл `.json` у запиті відсутній, відповідь буде лаконічною – лише кількість записаних значень.

Для видалення каналу за допомогою HTTP DELETE необхідно створити файл в форматі JSON з ідентифікатором користувача та передати його на дану адресу:

```
https://api.thingspeak.com/channels/<channel_id>
```

Запис даних в таблицю каналу відбувається за допомогою HTTP POST або HTTP GET запитів. Для передавання даних на сервер необхідно зробити запит з наступними посиланням:

```
https://api.thingspeak.com/update.json?api_key=<write_api_key>&field1=<data>
```

де `<write_api_key>` – ключ для запису в даний канал, `<data>` – дані які необхідно передати на сервер. У відповідь на запит буде отриманий JSON файл з датою та часом запиту, а також даними які були передані та ідентифікатором таблиці.

Наприклад, надіслати у створений для проектованої системи канал дані можна шляхом введення в адресному рядку браузера:

```
https://api.thingspeak.com/update.json?api_key=EI3MU9PVGK0HP4TW&field1=173
```

Читання даних з таблиці каналу відбувається за допомогою HTTP GET запита. Для читання даних з сервер необхідно зробити запит з наступними посиланням:

```
https://api.thingspeak.com/channels/<channel_id>/feeds.json
```

У відповідь на запит отримуємо JSON файл з усіма даними в цьому каналі. Для зменшення отримання даних можна додати параметр results та вказати скільки даних необхідно отримати, чим можна відкинути застарілі дані. Також можна відсортувати дані за значенням або часом подання.

Для читання останнього значення з таблиці можна використати GET запит:

```
https://api.thingspeak.com/channels/<channel_id>/feeds/last.<format>
```

де <format> – формат отриманих даних.

Для читання даних про налаштування певного каналу необхідно використати GET запит:

```
https://api.thingspeak.com/channels/<channel_id>.<format>
```

Для оновлення даних про певний канал використовується PUT запит, в якості параметрів якого передається JSON файл з налаштуваннями каналу. Для отримання даних про усі канали необхідно використати GET запит:

```
https://api.thingspeak.com/channels.json?api_key=XXXXXXXXXXXXXXXXXX
```

де параметру `api_key` присвоюємо значення ключа користувача. В результаті запиту буде отриманий файл з інформацією про всі канали користувача.

Для отримання часу від останнього запису необхідно виконати наступний API GET запит:

https://api.thingspeak.com/channels/<channel_id>/feeds/last_data_age.json

Ці запити можна використати при написанні програмного забезпечення для модуля Raspberry Pi проектованої системи, яке буде передавати дані від передавачів до хмарного серверу ThingSpeak.

3.3. Програмне забезпечення для системи контролю якості питної води

В проектованій системі програмне забезпечення складається з двох частин. Перша частина ПЗ призначена для управління мікроконтролером ATmega328P, який є ядром модуля Arduino Uno. Воно написане на мові програмування C++ в середовищі Arduino IDE. Друга частина ПЗ проектованого пристрою написана для RPi. Воно відповідає за отримання даних від Arduino, їх перетворення та передачу у віддалений хмарний сервер

Алгоритми роботи системи.

Блок-схема алгоритму програми для мікроконтролера модуля Arduino Uno наведена на рис. 3.7.

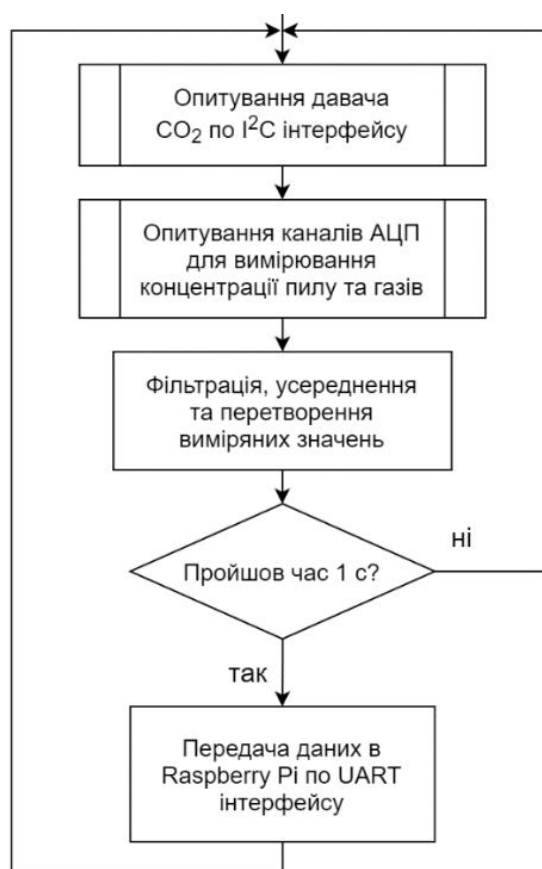


Рис. 3.7. Блок-схема алгоритму роботи програми для модуля Arduino Uno

Після ввімкнення пристрою здійснюється налаштування виводів мікроконтролера та ініціалізація інтерфейсів. Виміряні дані фільтруються і множаться на відповідні коефіцієнти для отримання реальних числових значень. На наступному етапі дані надсилаються в модуль RPi використовуючи послідовний інтерфейс. Таким чином, пристрій працює циклічно, до тих пір поки система не відключиться від джерела живлення. На рис. 3.8 зображена блок-схема алгоритму роботи програми для мікрокомп'ютера RPi.

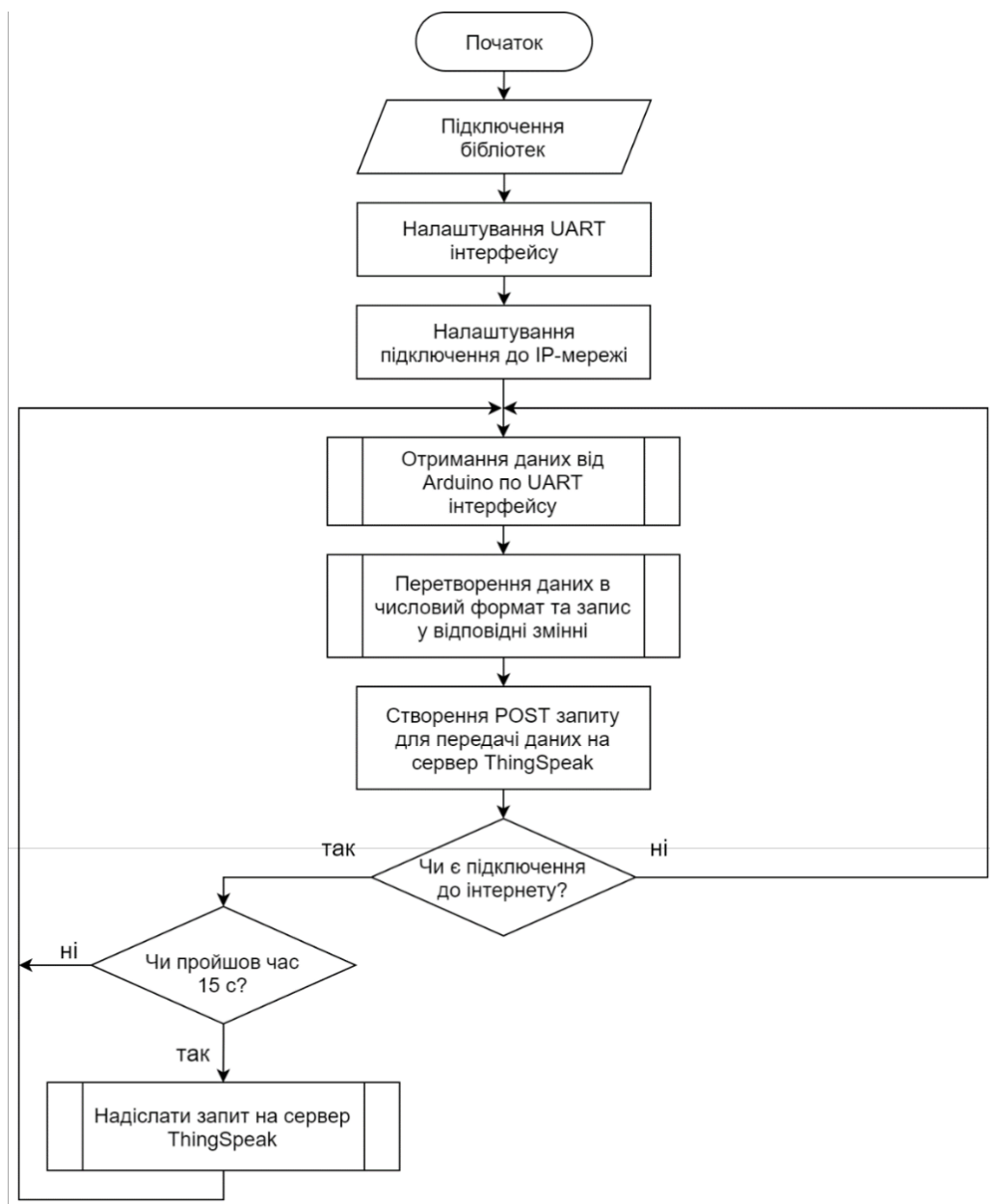


Рис. 3.8. Блок-схема алгоритму роботи програми для Raspberry Pi

На першому етапі здійснюється підключення бібліотек, налаштування UART інтерфейсу та ініціалізація змінних та констант. Після цього відбувається отримання даних від модуля Arduino та їх опрацювання.

Модуль Arduino надсилає повідомлення в послідовний порт у стрічковому форматі, після чого з нього витягуються числові значення. В Python найпростіше отримати його у вигляді рядка, а потім скористатися функцією `split()`, щоб розділити і представити як список значень. Для доступу до окремих елементів списку потрібно використовувати індекси в квадратних дужках. Наприклад, перший елемент списку має нульовий індекс, наступний – перший і т.д. Після цього відбувається перетворення стрічкового формату числа в цілочисельний тип, щоб в подальшому оперувати ним у програмі як числом.

Паралельно з цим, мікрокомп'ютер RPi намагається підключитися до мережі інтернет. Після успішного встановлення з'єднання відбувається передача даних на веб-сервер використовуючи методи API REST.

Опис програмних функцій та модулів.

При написанні коду для модуля Arduino Uno для реалізації функцій блоку моніторингу питної води було використано такі бібліотеки:

«I2C.h» для роботи з I²C інтерфейсом;

«UART.h» для роботи із UART інтерфейсом;

«MQ135.h» для роботи із датчиком.

Для отримання даних з послідовного порта в Raspberry Pi застосовуються програмні засоби Python і бібліотеки `pySerial`. На першому етапі відбувається налаштування підключення Raspberry Pi до мережі. Для сканування мереж Wi-Fi використовується команда `sudo iwlist wlan0 scan`. В результаті її виконання на екран виведуться всі доступні мережі WiFi, та характеристика даної точки доступу. Відкривши конфігураційний файл `wpa-supPLICANT` за допомогою команди:

```
sudo nano /etc/wpa_supPLICANT/wpa_supPLICANT.conf
```

Код для підключення RPi до точки доступу має вигляд (рис. 3.9).

```
network={
    ssid="Link"
    psk="Password"
}
```

Рис. 3.9. Лістинг коду для підключення RPi до точки доступу

Після цього файл може бути збережений. Для перевірки наявності підключення до WiFi використано команду `ifconfig wlan0`. Якщо поле `inet addr` містить адресу, то це означає, що Raspberry Pi успішно підключився до мережі.

Для простого налаштування та передавання даних на сервер ThingSpeak було використано бібліотеки мови Python `urllib` та `html.client`. Для встановлення бібліотек необхідно виконати наступну команду в терміналі ОС RPi.

```
sudo pip install urllib
sudo pip install html.client
```

Бібліотека `html.client` реалізує клієнтську частину протоколів HTTP та HTTPS та реалізує наступні класи `HTTPConnection`, `HTTPSConnection`, `HTTPResponse` та класи, які викликаються при некоректному підключенні. Використовуючи дану бібліотеку, можна передавати запити на сервер. Для обміну даними використовуються події API REST (POST, GET, PUT та DELETE), що дозволяє оновити існуючий канал, створити канал, очистити або видалити канал.

Також до проекту підключено `urllib` – це пакет, який об'єднує кілька модулів для роботи з URL-адресами:

- `urllib.request` – для відкриття та читання URL-адрес;
- `urllib.error` – містять винятки, які обробляють
- `urllib.parse` – для отримання URL-адрес;
- `urllib.robotparser` – для аналізу `robots.txt` файлів.

Усі необхідні бібліотеки були підключені до файлу програми (рис. 3.10).

```
import http.client
import urllib
from time import sleep, localtime, strftime
import time
import serial
```

Рис. 3.10. Лістинг коду для підключення бібліотек до файлу програми

Після підключення усіх необхідних бібліотек було реалізовано процес отримання даних з СОМ порта. Для цього потрібно створити змінні для усіх необхідних ключів доступу для API (рис. 3.11).

```
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
ser.flush()
write_api_key = 'EI3MU9PVGK0HP4TW'
```

Рис. 3.11. Лістинг коду для створення змінних для усіх ключів доступу для API

У циклі перевіряється чи Arduino передає дані в СОМ порт. Якщо так, то стрічка розшифровується. Отримані дані можна передавати, шляхом створення запиту (рис. 3.12).

```
line = ""
if ser.in_waiting > 0:
    line = ser.readline().decode('utf-8').rstrip()
    print(line)
    arr = line.split()
    params = urllib.parse.urlencode({'field1':'20','api_key':write_api_key})
    headers = {"Content-type":"application/x-www-form-urlencoded","Accept":
"text/plain"}
    conn = http.client.HTTPConnection("api.thingspeak.com:80")
```

Рис. 3.12. Лістинг коду для створення запиту при передачі даних в СОМ порт

Після цього передається POST запит на сервер з даними для оновлення та очікування відповіді (рис. 3.13).

```

conn.request("POST", "/update", params, headers)
response = conn.getresponse()
print(strftime("%a, %d %b %Y %H:%M:%S", localtime()))
print(response.status, response.reason)
data = response.read()
conn.close()

```

Рис. 3.13. Лістинг коду для передачі POST запиту на сервер для оновлення даних

Якщо запит на сервер відбувся правильно в результаті буде отриманий звіт про успішне передавання даних.

3.3. Результати роботи програми в додатку ThingSpeak

На рис. 3.14-3.16 представлені графіки, які відображають динаміку зміни результатів вимірювання датчиками параметрів якості води впродовж певного періоду.

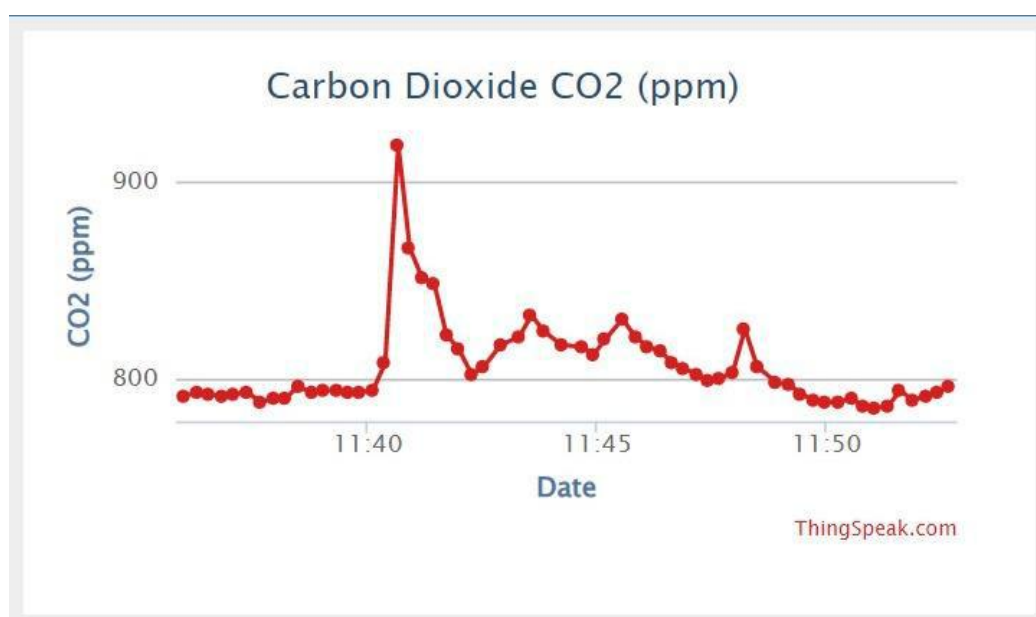


Рис. 3.14. Результат моніторингу концентрації карбону диоксиду

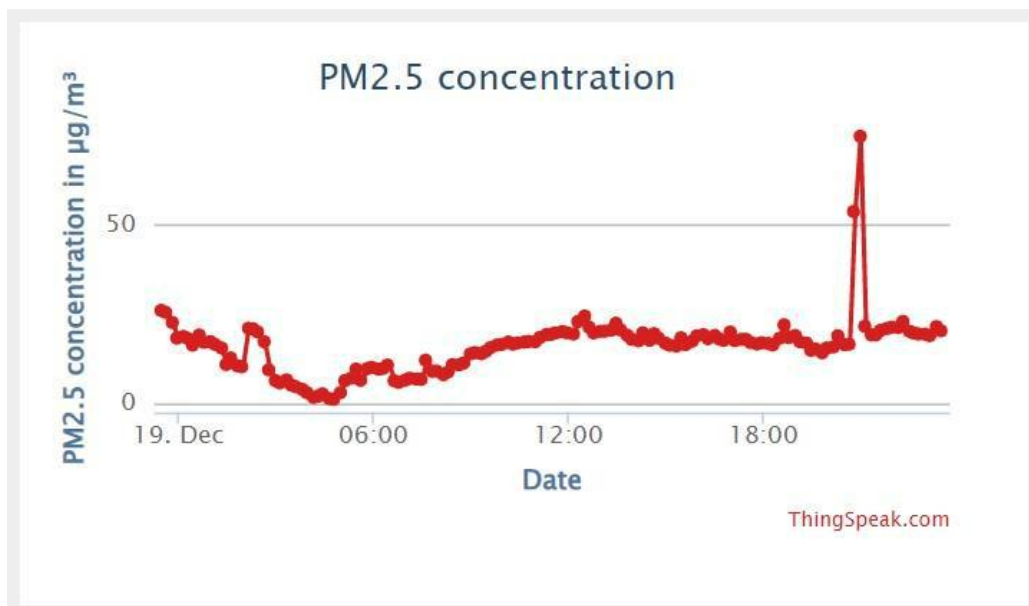


Рис. 3.15. Результат моніторингу концентрації PM2.5 у воді



Рис. 3.16. Результат моніторингу концентрації NH₃ у воді

Платформа ThingSpeak дозволяє зберігати накопичені дані на сервері впродовж року. Дані можна експортувати в будь-яку програму для подальшого опрацювання. Імпортовані дані можуть бути використані сторонніми програмами для аналізу та візуалізації. Це дає багато можливостей для статистичного опрацювання даних, виконання додаткових розрахунків та поглибленого аналізу отриманих даних.

Для експорту даних з платформи ThingSpeak можна скористатись одним з двох методів. Перший вивантажує всі дані відразу і лише у форматі CSV. Другий є гнучкішим, оскільки дозволяє налаштувати поля для експорту даних і пропонує кілька форматів на вибір: JSON, XML або CSV. Використання сервісу ThingSpeak дає можливість вбудовувати графіки на сторінки сайту з можливістю масштабування для створення власних систем моніторингу.

Збільшення вхідних потоків даних за рахунок підключення додаткових сенсорів до блоку моніторингу значно розширить можливості системи. Реалізація даної схеми можлива, оскільки модуль Arduino Uno має ще 3 вільних вхідних канали АЦП.

ВИСНОВКИ

У даній кваліфікаційній роботі була вдосконалена комп'ютеризована система для дистанційного моніторингу якості питної води.

Виконано огляд наукової літератури на тему кваліфікаційної роботи, проаналізовано сучасні розробки в сфері контролю якості питної води, визначено їх переваги і недоліки.

При вдосконаленні роботи комп'ютеризованої системи дистанційного моніторингу якості питної води було:

Написано код на мові програмування C++ для збору даних з датчиків, що вимірюють параметри якості питної води, і передавання їх у хмарний сервіс за допомогою технології IoT. Доведено, що система працює надійно. Реалізовано процес стабільного передавання вимірних даних на хмарний сервер за технологією REST API.

Визначенно цілі та завдання систем моніторингу якості питної води.

Проаналізовано систему моніторингу стану якості питної води.

Сформовано критерії вибору системи для моніторингу води.

Виконано аналіз комп'ютеризованої системи дистанційного моніторингу.

Вдосконалено комп'ютеризовану систему для дистанційного моніторингу якості питної води..

Впровадження вдосконаленої системи дозволить в режимі реального часу здійснювати контроль якості питної води, що дасть змогу покращити екологічні показники.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Real-time distance monitoring system URL: <https://drisk.a/real-time-monitoring-of-social-distancing/>
- 2.Revathy V. S., Ganesan K., Rohini K., ChindhuT. S., Boobalan T. Water QualityMonitoring System. Journal of Electronics and Communication Engineering. Vol.11. Issue 2. Ver. II. 2016. P. 27–40.
- 3.Басманов Є. І. Наука про воду: конспект, місто Харків, 2004, 76 с.
- 4.Мислива Т. М., Долгілевич М. Й. Основи моніторингу довкілля: навч. посібник. Житомир: Вид-во ДВНЗ «Державний агроєкологічний університет». 2007. 376 с.
- 5.Моніторинг поверхневих вод, Державне агентство водних ресурсів України, 2020, URL: <https://www.davr.gov.ua/monitoring-poverhnevih-vod1> (дата звертання (28.04.2023))
- 6.Живага В. В., Малахова М. О., Шевченко Д. О. Система моніторингу якості води на базі IoT. Вісник Харківського національного університету імені В.Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». Вип. 49. 2021. С. 49-57.
- 7.Бахарєв В. С., Маренич А. В. Аналітичний огляд результатів наукових досліджень з проблем моніторингу довкілля в Україні. Екологічна безпека, № 2. 2016. С. 35-42./
8. Chaudhryl V. Arduair: Water Quality Monitoring. International Journal of Environmental Engineering and Management. 2013. P. 639-646.
- 9.Zhanga H., Wangb Sh., Haob J., Wangc X., Wanga Sh., Chaia F., Lid M. Water quality and control action in Beijing. Journal of Cleaner Production. Vol. 112. Part 2. 2016. P. 1519-1527.
- 10.JunHo J., ByungWan J., JungHoon K., SungJun K., WoonYong H. Development of an IoT-Based Indoor Water Quality Monitoring Platform. Journal of Sensors. Vol. 2020. P. 1-14.

11. Okokpuije K., Noma-Osaghae E., Modupe O., John S., Oluwatosin O. A smart water pollution monitoring system. *International Journal of Civil Engineering and Technology*. Vol. 9. 2018. P. 799–809.
12. Іванюк О. О., Влах-Вигриновська Г. І., Близнюк А. М., Сапіга І. В. Система виявлення та очищення води від шкідливих речовин на базі технології інтернету речей. *Автоматика, вимірювання та керування*. Т. 2, № 1 (2). 2020. С. 40-48.
13. Parmar G., Lakhani S., Chattopadhyay M. An IoT based low cost water quality monitoring system. *International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*. 2017. P. 524-528.
14. Путренко В. В., Назаренко С. Ю. Визначення якості питної води на основі інтелектуального аналізу даних. *Математичне моделювання в економіці*. №3-4. 2016. С. 176-187.
15. SHAH, Joy. Model for water monitoring quality distribution., 2017, 6.3: 3446-3451.
16. Zhao Y. L., Tang J., Huang H. P., Wang Z., Chen T. L., Chiang C. W., Chiang P. C. Development of IoT Technologies for Water Prevention and Improvement. *Aerosol and Water Quality Research*. 2020. P. 2874-2888.
17. Мокін В. Б., Собко Б. Ю., Дратований М. В., Крижановський Є. М., Горячев Г. В. Створення інформаційної системи моніторингу води міста на основі технології Інтернет речей. *Вісник Вінницького політехнічного інституту*. № 3. 2017. С. 49-58.
18. ManicTime. Time Tracking with Automatic Work Hour Logging. URL: <https://www.manictime.com/>
19. OfficeTime. Time and Expense Tracking Software. URL: <https://www.officetime.net/>
20. actiTIME. Time Tracking Software for Businesses. URL: <https://www.actitime.com/>
21. Chrometa. Automatic Time Tracking Software. URL: <https://www.chrometa.com/>
22. Paymo. Work and Project Management Software. URL: <https://www.paymoapp.com/>

23. MyHours. Time Tracking for Freelancers and Teams. URL: <https://myhours.com/>
24. Hubstaff Tasks. Agile Project Management. URL: <https://hubstaff.com/tasks>
25. Insightful (formerly Workpuls). Employee Monitoring and Productivity Insights. URL: <https://www.insightful.io/>
26. Connecteam. Employee Management App with Time Tracking. URL: <https://www.connecteam.com/>
27. Jibble. Free Time and Attendance Tracking Software. URL: <https://www.jibble.io/>
28. OnTheClock. Employee Time Clock and Attendance Software. URL: <https://www.ontheclock.com/>
29. FunctionFox. Timesheets and Project Management Software. URL: <https://www.functionfox.com/>
30. Replicon. Time and Attendance Software. URL: <https://www.replicon.com/>
- Kranz Maciej. Building the Internet of Things: Implement New Business Models Disrupt Competitors Transform Your Industry. Wiley, 2016. 273 с.3.
33. Torben Lund Skovhus, Bo Højris Microbiological Sensors for the Drinking Water Industry. IWA Publishing, 2018. 334 с.
34. _Моніторинг якості води, атмосферного повітря, ґрунту
<https://oolc.od.ua/>
35. Мокін В. Б. Комп'ютеризовані регіональні системи державного моніторингу поверхневих вод: моделі, алгоритми, програми. / Мокін В. Б. — Вінниця: УНІВЕРСУМ — Вінниця, 2005. — 315 с.
36. Sumanta Guha Computer Graphics Through OpenGL / Sumanta Guha — University of Michigan: A K Peters/CRC Press, 2014. — 951 с.
38. Steve Marschne Fundamentals of Computer Graphics / Steve Marschner, Peter Shirley — Cornell University: A K Peters/CRC Press, 2015. — 734 с.
39. . Кукурудза С. І. Аналіз якості природних вод / С. І. Кукурудза , С. М. Турій — Львів.: Вид-во ЛДУ, 1994. — 37 с
40. Bartram, J. Water Quality Monitoring. /J. Bartram, R. Balance — London: E&FN Spon, 1996. — 400 с

ДОДАТКИ

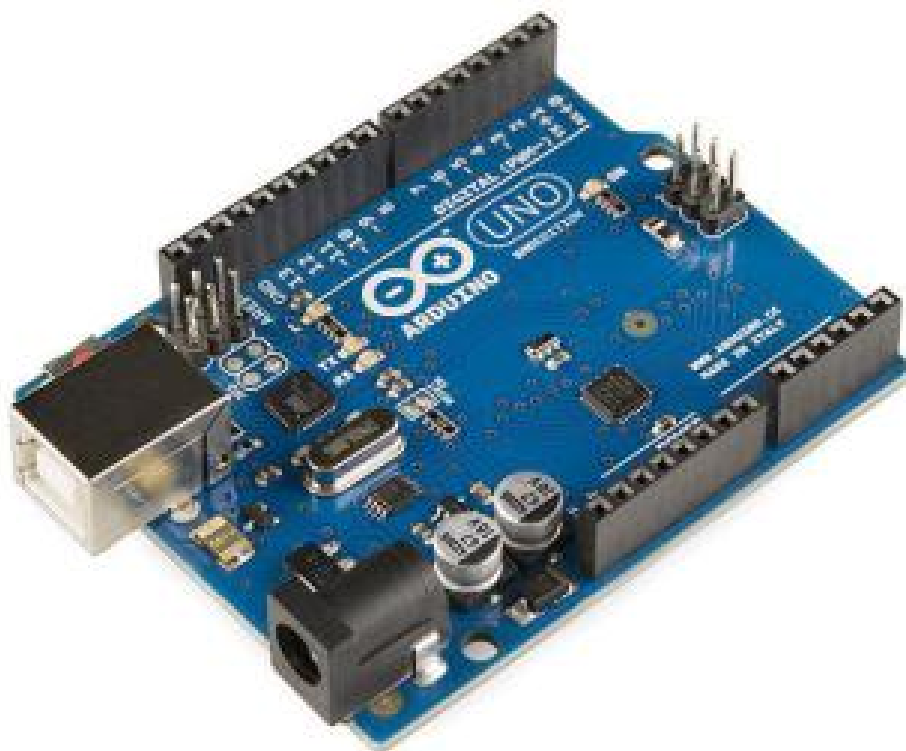
Додаток А

Функціональне призначення виводів RPi



Wedge Silk	Python (BCM)	WiringPi GPIO	Name	P1 Pin Number		Name	WiringPi GPIO	Python (BCM)	Wedge Silk
			3.3v DC Power	1	2	5v DC Power			
SDA		8	GPIO02 (SDA1, I2C)	3	4	5v DC Power			
SCL		9	GPIO03 (SCL1, I2C)	5	6	Ground			
G4	4	7	GPIO04 (GPIO_GCLK)	7	8	GPIO14 (TXD0)	15		TXO
			Ground	9	10	GPIO15 (RXD0)	16		RXI
G17	17	0	GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)	1	18	G18
G27	27	2	GPIO27 (GPIO_GEN2)	13	14	Ground			
G22	22	3	GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)	4	23	G23
			3.3v DC Power	17	18	GPIO24 (GPIO_GEN5)	5	24	G24
MOSI		12	GPIO10 (SPI_MOSI)	19	20	Ground			
MISO		13	GPIO09 (SPI_MISO)	21	22	GPIO25 (GPIO_GEN6)	6	25	G25
		(no worky 14)	GPIO11 (SPI_CLK)	23	24	GPIO08 (SPI_CE0_N)	10		CD0
			Ground	25	26	GPIO07 (SPI_CE1_N)	11		CE1
IDSD		30	ID_SD (I2C ID EEPROM)	27	28	ID_SC (I2C ID EEPROM)	31		IDSC
G05	5	21	GPIO05	29	30	Ground			
G6	6	22	GPIO06	31	32	GPIO12	26	12	G12
G13	13	23	GPIO13	33	34	Ground			
G19	19	24	GPIO19	35	36	GPIO16	27	16	G16
G26	26	25	GPIO26	37	38	GPIO20	28	20	G20
			Ground	39	40	GPIO21	29	21	G21

Зовнішній вигляд модуля Arduino Uno



Електрична схема модуля Sensor W-25011 датчика

