МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Розробка застосунків інтернету речей та сенсорних мереж в енергетиці

Рекомендації до виконання курсової роботи

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського,
як навчальний посібник для здобувачів ступеня магістра
за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці»
спеціальності 121 Інженерія програмного забезпечення

Укладачі: Н. В. Федорова, Г. В. Сарибога, І. І. Гусєва

Електронне мережеве навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2024

УДК 004.6
Ф33

Укладачі: *Федорова Наталія Володимирівна, д.т.н., доцент*
Сарибога Ганна Володимирівна, старший викладач
Гусєва Ірина Ігорівна, к.е.н., доцент

Рецензент: *Жураковський Б.Ю., д-р. техн. наук, проф. професор кафедри інформаційних систем та технологій факультету інформатики та обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»*

Відповідальний редактор: *Недашківський О. Л., д.т.н., доцент*

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 2 від 8.11.2024 р.)
за поданням Вченої ради НН ІАТЕ КПІ ім. Ігоря Сікорського
(протокол № 4 від 28.10.2024 р.)

Ф33 **Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці.** Рекомендації до виконання курсової роботи [Електронний ресурс]: навч. посіб. для здобувачів ступеня магістра за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського; уклад.: Н.В. Федорова, Г.В. Сарибога, І.І. Гусєва; – Електронні текстові дані (1 файл: 2,13 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2024. – 37 с.

Навчальний посібник призначено для організації виконання курсової роботи, підготовки, оформлення та захисту курсової роботи з дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці». Посібник містить організаційні вказівки до виконання курсової роботи, детальні вимоги до її оформлювання, а також теоретичні відомості та практичні рекомендації щодо розробки програмного забезпечення відповідно до завдання.

УДК 004.6

Реєстр. № 24/25-110. Обсяг 1,68 авт. арк.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Берестейський, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

ЗМІСТ

ВСТУП

Курсова робота з дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці» є науково-практичною роботою студентів, що охоплює основні теми даної дисципліни [1].

Метою виконання курсової роботи є перевірка засвоєння теоретичних знань студента з дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці» та вміння їх практичного застосування [2].

Виконання курсової роботи спрямовано на поглиблення знань з певних тем дисципліни, отримання навичок з аналізу практичних питань матеріалу, що вивчається в дисципліні, а також вивчення навчально-наукової літератури та на перевірку засвоєння знань студентами з питань розробки застосунків Інтернету речей та сенсорних мереж в енергетиці.

Успішне виконання курсової роботи є показником:

- поглибленого засвоєння студентом матеріалу, що вивчається в дисципліні «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці»;
- вміння працювати з літературними джерелами, а саме: монографічними, навчальними, загальними, спеціальними, а також нормативно-правовими документами та матеріалами практичних досліджень;
- здібностей студента до самостійного аналізу проблемних питань з розробки застосунків Інтернету речей та сенсорних мереж в енергетиці;
- вміння письмово викладати результати своєї роботи.

Для успішного виконання одного з основних видів письмових робіт - курсової роботи, пропонується до уваги даний навчальний посібник, який може бути використаний також при написанні дипломної роботи.

Навчальний посібник містить вимоги, що спрямовані на підвищення якості курсової роботи, а також передбачає вимоги, які мають бути обов'язково виконані студентами та рекомендації, реалізувати які доцільно для підвищення якості курсових робіт.

Студентам для написання курсової роботи пропонуються на вибір шість тем, які складаються з ряду підтем, в рамках вивчення дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці». Тема вибирається студентом самостійно.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Метою дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці» є набуття знань та практичних навичок використання сучасних методів щодо розробки застосунків Інтернету речей та сенсорних мереж в енергетиці.

Предметом дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж» є програмне забезпечення, що спрямоване на розробку застосунків Інтернету речей та сенсорних мереж в енергетиці з використанням сучасних методів, підходів та інструментів.

Основні завдання кредитного модуля.

Згідно з вимогами програми навчальної дисципліни студенти після засвоєння кредитного модуля мають продемонструвати такі результати навчання [3]:

Фахові компетентності:

- здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення (ФК 1);
- здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення (ФК 2);
- здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів (ФК 3);
- здатність забезпечувати якість програмного забезпечення (ФК 9);
- здатність розробляти програмні застосунки інтернету речей та сенсорних мереж (ФК 10).

Згідно з вимогами ОПП/ОНП Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці, студенти після засвоєння навчальної дисципліни мають продемонструвати такі результати навчання.

Програмні результати навчання [3]:

- оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу (ПРН 2);
- виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення (ПРН 4);
- розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення (ПРН 5);
- забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення (ПРН 11);

- конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу (ПРН 13);

- планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення (ПРН 16);

- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела (ПРН 17);

- вміти розробляти програмні застосунки інтернету речей та сенсорних мереж (ПРН 18).

Дисципліна складається з одного кредиту [2, 3].

Вивчення дисципліни спирається на знання, отримані за програмою попередніх років навчання за спеціальністю 121 Інженерія програмного забезпечення. Студенти мають досвід у імперативному, об'єктно-орієнтованому і функціональному програмуванні.

Викладений матеріал може бути інструментальною основою для підготовки магістерських дисертацій.

Міждисциплінарні зв'язки забезпечуються дисциплінами «Аналітика обробки даних в сенсорних мережах», «BigData в енергетиці» «Математичні методи моделювання систем з розподіленими параметрами», «Розробка програмного забезпечення мобільних пристроїв» [2, 3].

2. ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

2.1. Етапи виконання курсової роботи

Основні етапи та орієнтовний графік виконання курсової роботи подано в табл. 2.1.

Таблиця 2.1.

Основні етапи та орієнтовний графік виконання курсової роботи

Тиждень семестру	**Назва етапу роботи**	**Навчальний час**
		СРС
1-2	Отримання теми та завдання. Підбор та вивчення літератури	2
3	Розробка концепції системи моніторингу	2
4-5	Проведення аналізу: - існуючих датчиків; - вибору платформи обробки великих даних (Big Data); - вибір мови програмування	2
6-7	Написання скрипта/програми щодо збору даних з датчиків	4
8-9	Імітація навантаження (потоків даних) – симуляція навантаження з датчиків (Big Data)	2
10-11	Застосування методів обробки Big Data	4
12-13	Візуалізація даних	4
14-15	Написання пояснювальної записки та підготовка презентації	6
16	Подання курсової роботи на перевірку	2
17	Захист курсової роботи	2
	Всього годин	30

2.2. Перелік завдань для виконання курсової роботи

Студент обирає тему курсової роботи самостійно з тем, що запропоновані в даному навчальному посібнику. Самовільна зміна варіанта або теми курсової роботи не допускається. Зміна варіанта або назви теми можлива тільки при узгодженні з викладачем.

В даній курсовій роботі необхідно розробити [3]:

- систему моніторингу показників електролічильників;
- систему моніторингу енергобезпеки в будівлі;
- систему моніторингу енергоспоживання в електроустановках;
- систему моніторингу для сонячної енергії;
- систему енергетичного моніторингу;
- систему моніторингу енергоефективності будівель.

Кожна з тем включає в себе по 5 підтем чи етапів виконання роботи, які тісно пов’язані між собою та одночасне виконанні яких формують тему курсової роботу в цілому. Завдання на курсову роботу спрямоване на розвиток навичок роботи в команді [3].

Тема 1. Розробка системи моніторингу показників електролічильників

Етапи роботи (загальна робоча група - 14 студентів):

1. Проведення аналізу - 2 студенти:
    a. існуючих датчиків;
    b. вибору платформи обробки великих даних;
    c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з лічильників – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з лічильників - 3 студенти.
4. Застосування методів обробки Big Data - 3 студенти.
5. Візуалізація даних - 3 студенти.

Тема 2. Розробка системи моніторингу енергобезпеки в будівлі

Етапи роботи (загальна робоча група - 14 студентів):

1. Проведення аналізу - 2 студенти:
    a. існуючих датчиків;
    b. вибору платформи обробки великих даних;
    c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з датчиків – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків - 3 студенти.
4. Застосування методів обробки Big Data - 3 студенти.
5. Візуалізація даних - 3 студенти.

Тема 3. Розробка системи моніторингу енергоспоживання побутових приладів

Етапи роботи (загальна робоча група -14 студентів):

1. Проведення аналізу – 2 студенти:
    a. існуючих датчиків;
    b. вибору платформи обробки великих даних;
    c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з датчиків – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків - 3 студенти.
4. Застосування методів обробки Big Data – 3 студенти.

5. Візуалізація даних – 3 студенти.

Тема 4. Розробка системи моніторингу для сонячної енергії

Етапи роботи (загальна робоча група - 14 студентів):

1. Проведення аналізу - 2 студенти:
   a. існуючих датчиків;
   b. вибору платформи обробки великих даних;
   c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з датчиків – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків - 3 студенти.
4. Застосування методів обробки Big Data - 3 студенти.
5. Візуалізація даних - 3 студенти.

Тема 5. Розробка системи енергетичного моніторингу

Етапи роботи (загальна робоча група - 14 студентів):

1. Проведення аналізу - 2 студенти:
   a. існуючих датчиків;
   b. вибору платформи обробки великих даних;
   c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з датчиків – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків - 3 студенти.
4. Застосування методів обробки Big Data - 3 студенти.
5. Візуалізація даних - 3 студенти.

Тема 6. Розробка системи моніторингу енергоефективності будівель

Етапи роботи (загальна робоча група - 14 студентів):

1. Проведення аналізу - 2 студенти:
   a. існуючих датчиків;
   b. вибору платформи обробки великих даних;
   c. вибір мови програмування.
2. Написання скрипта (програми) щодо збору даних з датчиків – 3 студенти.
3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків - 3 студенти.
4. Застосування методів обробки Big Data - 3 студенти.
5. Візуалізація даних - 3 студенти.

3 ТЕОРЕТИЧНІ ОСНОВИ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

3.1. Концепція Інтернету речей

Вперше термін та поняття «Інтернету речей» озвучив американський фахівець з телекомунікацій Пітер Льюїс у 1985 році. Він представив IoT як «інтеграцію людей, процесів і технологій з пристроями для підключення і датчиками для забезпечення віддаленого моніторингу, стану, маніпулювання та оцінки тенденцій таких пристроїв» [4] .

Для IoT потрібні дешеві, енергозберігаючі та досить потужні процесори, які можна масово штампувати на заводах. Цим проривом став винахід RFID, не менш важливим кроком вважається прийняття стандарту IPv6, здатного забезпечити достатню кількість IP-адрес.

3.1.1. Засоби ідентифікації

IoT об'єднує мільярди пристроїв, тому їхня працездатність багато в чому залежить від того, наскільки добре машини розпізнають одне одного та наскільки їх розпізнають сервери та маршрутизатори всередині мережі, а також зовнішні хмарні системи управління. Зашифрований канал автентифікації необхідний, щоб кожен об'єкт отримав власний унікальний знак і міг підтвердити свою справжність. Заради безпеки системи повинні приймати дані та виділяти виробничі ресурси пристроям Інтернету речей тільки після авторизації останніх, адже це захищає від зламу та так званого спуфінгу — підміни даних.

У деяких промислових системах вже розгорнуті сервіси, де кожному пристрою надають універсальний унікальний ідентифікатор (UUID), який є 128-бітовим номером. Він містить мережеву адресу системи, що згенерувала UUID, мітку із зазначенням часу та випадково створений компонент, перших двох достатньо для гарантії унікальності, а випадковий елемент є додатковим захистом. Імовірність дублювання ідентифікатора можлива, але практично дорівнює нулю. Завдяки UUID потрібну річ можна легко знайти і не переплутати з іншими, тому ця технологія важлива для роботи IoT (рис.3.1).

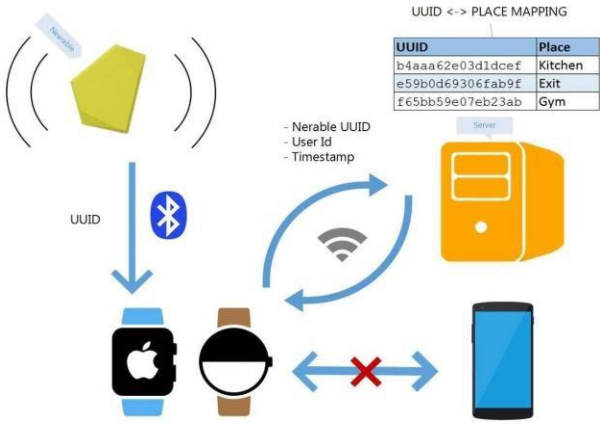

Рис. 3.1. Приклад системи взаємодії пристроїв IoT з використанням UUID

Згідно з дослідженням, опублікованим у журналі [5], поки не існує загальної схеми ідентифікації для IoT. Поки відсутність єдиного механізму ідентифікації ускладнює взаємодію між різними складовими Інтернету речей. Втім, деякі міжнародні організації надають свої стандарти та рішення для забезпечення безпеки IoT, наприклад, стандарти OCF - Open Connectivity Foundation, FIWARE, oneM2M. Тут же став у нагоді і глобальний інтернет-протокол IPv6, що забезпечує унікальні IP-адреси машин.

3.1.2. Засоби вимірювання

Елементи IoT використовують різні види датчиків для вимірювання та оцінки навколишнього середовища. Вони бувають різних розмірів і форм, деякі містять в собі менші датчики, щоб збирати кілька типів даних.

В [6] наведено 10 найпопулярніших типів датчиків, а саме:

Температурні датчики. Уловлюють коливання температури, включаючи різкі перепади, небезпечні на деяких видах виробництва.

Датчики вологості. Вимірюють кількість водяної пари в атмосфері, часто застосовуються в система вентиляції, кондиціювання та опалення, вони також допомагають передбачати погоду.

Датчик тиску. Слідкує за тиском у газах та рідинах і попереджає про різкі зміни, існують моделі, що фіксують витікання.

Сенсори руху. Служать для виявлення об'єктів з відривом, зазвичай, створюють електромагнітні поля чи вловлюють інфрачервоне випромінювання.

Датчики рівня речовин. Визначають наповненість ємностей рідинами, порошками та сипучими матеріалами, а також наявність відходів у сміттєвих баків.

Сенсори газу. Оцінюють якість повітря, визначають наявність токсичних, горючих чи небезпечних речовин.

Оптичні датчики. Перетворюють світло, що надходить в електричні сигнали, можуть використовуватися для розпізнавання об'єктів за допомогою камер спостереження.

Інфрачервоні датчики. Випромінюють або вловлюють промені в інфрачервоному спектрі, щоб вивчати характеристики навколишнього середовища, такі як температура об'єктів.

Акселерометри. Визначають зміну швидкості об'єктів, а також гравітації, які можуть попереджати про рух предметів, які не можна переміщати.

Гіроскопи. Вимірюють швидкість обертання об'єктів навколо своєї осі або тремтіння камери.

3.1.3. Засоби передачі даних

Головний консультант компанії "SNT Україна" Ярослав Боцман у своїй статті [7] написав, що донедавна IoT використовував традиційні технології бездротової передачі даних, такі як WiFi та 3G. Застосування радіотехнологій сильно ускладнює проблема живлення, адже не кожен датчик можна підключити до електромережі, тому мережі стандартів 802.11 стають менш популярними. Пристрої можна підключати і за допомогою проводів, наприклад до ліній електропередач, але це не завжди зручно і знижує гнучкість Інтернету речей.

Для підключення розумних пристроїв на відстані до 10 м застосовуються стандарти, що використовують mesh-архітектуру, де кожен вузол з'єднаний з іншим подібно до павутини і споживає мінімум енергії: 6LoWPAN, Bluetooth Low Energy (BLE), ZigBee IP і т.д. Для зв'язку великих відстані інженери розробляють спеціальні технології з низьким енергоспоживанням. На вертикальних ринках вже використовується низка технологій — C-UNB (Cooperative Ultra Narrowband), LoRa (Long Range), але найперспективнішими для Інтернету речей Ярослав Боцман назвав технології EC-GSM (Extended Coverage GSM) та NB-CIoT (Narrowband Cellular IoT), що використовують мережі стільникового зв'язку. Розробники пропонують виділити для них смуги частот нижче використовуваних у мобільних мережах, а операторам достатньо поставити на базових станціях відповідне обладнання та оновити програмне забезпечення для обміну даними.

3.1.4. Засоби обробки даних

Пристрої Інтернету речей часто не можна підключити до мереж з високою швидкістю передачі даних та низькою затримкою, хоча потужність дозволяє їм обробляти дані в межах "туманних обчислень" — це проміжний етап між дата-центрами та хмарними сервісами. Для цього їм необхідні мікроконтролери, які об'єднуються у розподілену мережу для зберігання та обробки даних, а також надають обчислювальні потужності для керування системами без участі людини. Вже структуровану інформацію передають на "хмару" через спеціальні інтерфейси (API), де їх продовжують обробляти, зокрема вручну [8].

Інструменти Інтернету речей генерують велику кількість даних [9], отриманих з датчиків та пристроїв, а потім збирають їх у централізованому сховищі інформації. Величезний обсяг необроблених даних зберігаються поряд із структурованою інформацією, такою як записи транзакцій та профілі клієнтів. Аналітичні платформи можуть використовувати їх для створення звітів та інфографіки (рис. 3.2). Інструменти аналітики на основі штучного інтелекту користуються Інтернетом речей. Передаючи великі обсяги різних даних, зібраних датчиками IoT, можна ефективно навчати і обробляти інформацію та складати прогнози у реальному часі.

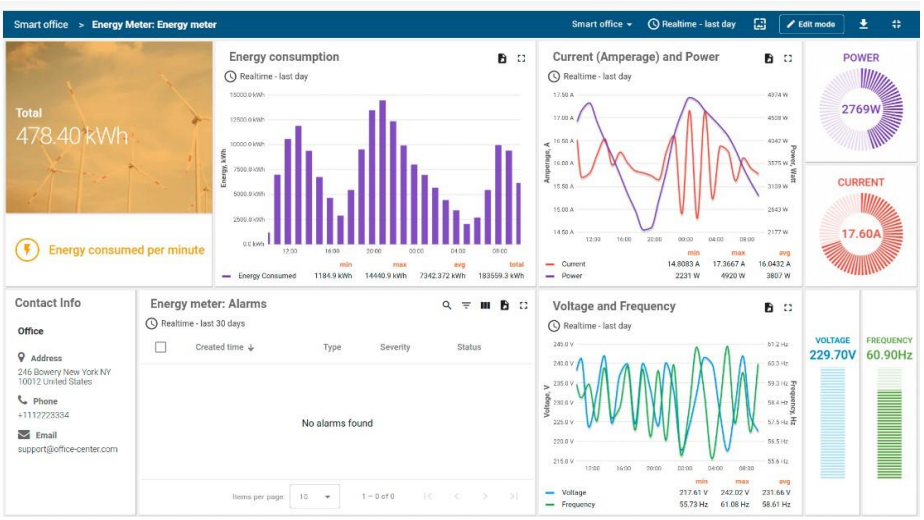

Рис. 3.2. Приклад інфографіки

3.1.5. Проблема безпеки IoT

Аналітики з компанії Apriorit дійшли висновку: IoT схильний приблизно до тих самих ризиків, що і звичайний інтернет:

- вразливість програмного забезпечення;
- незахищені канали зв'язку;
- витік з інформаційних систем;
- шкідливі програми;
- кібератаки.

Загроза зламу з боку зловмисників може відлякати багатьох клієнтів від використання Інтернету речей. Особливо слід побоюватися організаціям, які працюють у сфері медицини, фінансів, логістики, торгівлі та виробництва, адже будь-який злам може завдати шкоди тисячам людей. Серед іншого, пристрої IoT збирають конфіденційні дані людей, витік яких призведе до плачевних наслідків. Згідно з даними SAM Seamless Network, у 2021 році здійснено понад 1 млрд атак на мережі IoT.

Впровадити механізми захисту у системи IoT складніше, ніж у звичайні гаджети із підключенням до інтернету. По-перше, маленькі датчики та мікропроцесори складніше захистити на апаратному рівні, по-друге, інструменти безпеки збільшать вартість і час, що витрачаються на їх виробництво, тоді як для великої кількості пристроїв важлива саме дешевизна.

3.2. IoE-Інтернет енергії

Internet of Energy (IoE) є впровадженням технології Інтернету речей (IoT) у розподілені енергетичні системи. Інтернет енергії будується на принципах Інтернету речей , щоб надавати людям дані, необхідні для оптимізації та управління електромережею [10]. Мета - підвищити автономність роботи електромережі .

Використовуючи пристрої IoT , такі як інтелектуальні датчики та комунікаційні технології, енергетична галузь створює Internet of Energy для управління виробництвом енергії та енергетичними ресурсами .

Впровадження Internet of Things в електроенергетику не обмежується традиційним ринком побутового та промислового електроспоживання. IoE також пов’язаний з екологічним порядком денним: «зеленими» видами електроенергії й електротранспортом. На вітряних або сонячних станціях співробітник спостерігає за великою кількістю однакових електроблоків. Платформа IoT спрощує й систематизує цей процес, налагоджує роботу і збільшує ефективність станції.

Smart metering

Smart metering — це система розумних датчиків, один із сервісів Smart city [11]. Використовується для віддаленого моніторингу та контролю показань лічильників електроенергії, водо- та газопостачання, обліку теплової енергії та для своєчасного виявлення аварійних ситуацій. Розумний лічильник допомагає правильно розрахувати витрати за комунальні послуги та уникнути зайвих витрат для споживача.

У рішеннях для розумного міста використовується телеметрія — комплекс технологій, за допомогою яких можна контролювати різні об’єкти та процеси на відстані. Вона дозволяє дистанційно збирати та обробляти дані, спостерігати за станом мереж, наприклад, у багатоквартирних та приватних будинках. Зокрема, GSM-телеметрію використовують у промисловості: підприємства та заводи з її допомогою відстежують обсяг споживання ресурсів, а також стан власних транспортних магістралей.

Розумні системи обліку мають широку сферу застосування, у чому полягає одна з їхніх переваг (рис.3.3).

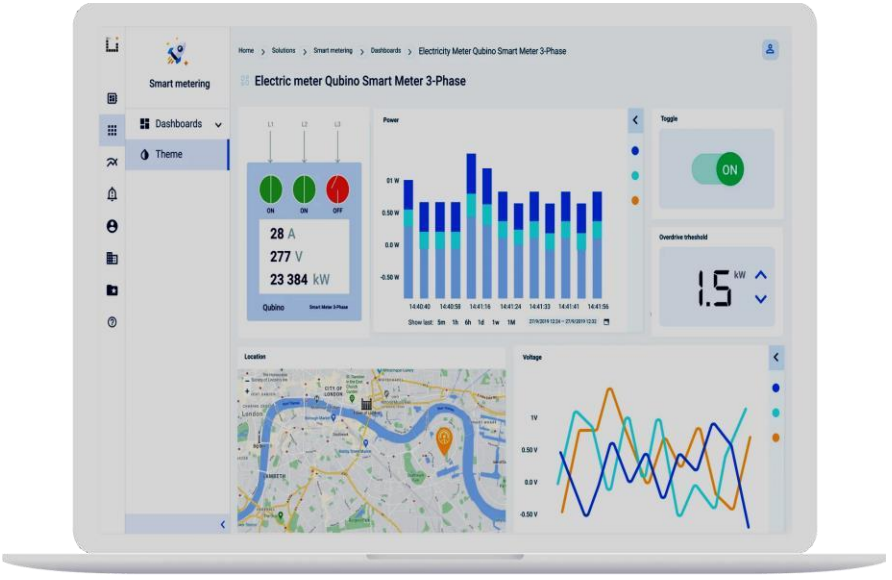

Рис. 3.3. Аналітика даних показників Smart meter

Вони актуальні для будь-якого енергоносія:

- електроенергія;
- вода;
- тепло;
- газ;
- нафта.

Автоматизації підлягають системи водопостачання та енергетики як житлових будинків, так і системи на підприємствах. Це вигідне рішення для промисловості, оскільки розумні прилади обліку дозволяють контролювати споживання ресурсів та стежити за справністю мереж.

Таким чином, розумні лічильники не тільки допомагають економно споживати ресурси, а й дозволяють мінімізувати наслідки аварій. Економія та безпека — пріоритети для кожного споживача.

Жителі багатоквартирних будинків отримують низку переваг, використовуючи розумні лічильники. Якщо датчики встановлені у системі водопостачання, можна уникнути зайвих витрат за оплату комунальних послуг. Зі Smart metering показання завжди точні та передаються автоматично, без необхідності контролю людиною [11].

Автоматизована система дозволяє підрахувати об’єм води, використаний конкретним будинком. Потім обчислюється обсяг, який використовували мешканці будинку за лічильниками. Сума, що залишилася, ділиться на всіх жителів, у яких немає лічильників. Завдяки цьому можна скоротити витрати та не платити за воду, яку ви не використовували.

Встановлення розумних лічильників входить до автоматизованої системи обліку, яка отримує та передає показання, наприклад, про використання води. На основі цих даних система будує модель споживання та може сигналізувати про відхилення. Якщо датчики встановлені в магістралі, то повідомлення про падіння тиску внаслідок прориву труби приходить до центру управління та про це негайно дізнається аварійна бригада. Завдяки своєчасним повідомленням комунальні служби можуть контролювати стан мереж та мінімізувати витрати від аварій.

Система розумних датчиків працює на основі інформаційних та комунікаційних технологій у поєднанні з IoT-технологіями (Інтернетом речей). Також використовується спеціалізоване програмне забезпечення, яке розроблене для розумного міста та його складових. Для телеметрії використовується стандарт NB-IoT [12], який застосовується для пристроїв із низькими обсягами обміну даними. Його переваги полягають у гнучкому управлінні енергоспоживанням, великою ємністю мережі.

Також варто зазначити, що сигнал може проникати навіть у підвальні приміщення з товстим залізобетонним фундаментом.

NB-IoT-технології забезпечують надійне, захищене від перешкод з’єднання з мережею. Для передачі даних використовуються частоти, що ліцензуються — такі канали зв’язку контролюються державною службою від незаконного втручання.

Технологія може бути реалізована у двох напрямках [12]:

- з новими лічильниками;
- зі старими лічильниками.

Лічильники старого зразка вимагають заміни із встановленням спеціального обладнання. Усі складові, включаючи новий лічильник, розміщуються в габаритній шафі вагою близько 130 кг, яка надійно захищає конструкцію. Перш ніж підключити таку конструкцію до будинку, проводиться експертна оцінка та складається індивідуальний план підключення.

Якщо вже встановлені імпульсні лічильники нового зразка або лічильники з виходом wireless M-Bus, залишається лише підключити спеціальні модеми, які зчитують інформацію та передають її NB-IoT-мережі в хмарну платформу [12].

Дані від вузла обліку проходять за допомогою спеціального програмного забезпечення, потрапляють у систему управління підприємством відповідного водоканалу. Для дистанційного керування SIM-картами та їх трафіком використовується Центр керування IoT від Київстар. Цей сервіс дозволяє автоматизувати керування IoT SIM-картками та налаштувати віддалений контроль за частотою та обсягами передачі даних у режимі 24/7.

4. ВИМОГИ ДО ЗМІСТУ ТА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

4.1. Вимоги до змісту курсової роботи

Курсова робота повинна відповідати таким вимогам [13]:

- точність у визначеннях та класифікаціях;
- чіткість побудови;
- логічна послідовність викладу матеріалу;
- аргументація висновків з кожного питання плану;
- конкретність у викладі результатів роботи;
- доведеність висновків щодо роботи в цілому;
- всі розділи курсової роботи мають бути логічно пов'язані між собою.

Курсова робота повинна містити:

- титульний лист (Додаток А);
- зміст із точним зазначенням назв розділів (підрозділів) та сторінок (Додаток Б);
- перелік умовних позначень, символів і скорочень (Додаток В);
- вступ (1 – 1,5 сторінки), в якому виявляється актуальність та значущість теми, робиться короткий огляд (аналіз) літератури (Додаток Г);
- основна частина складається з розділів, розбитих на підрозділи. Усі вони повинні мати заголовки, які не повторюють загальну назву роботи. Кожен розділ (підрозділ) завершується короткими висновками, які допоможуть зробити узагальнення ув'язнення (Додаток Д);
- висновок (1,5 – 2 сторінки), у якому підбиваються підсумки, робляться основні висновки, даються практичні рекомендації (якщо цього вимагає тема) (Додаток Е);
- список використаних джерел, що наводиться наприкінці курсової роботи (Додаток Ж) та оформлюється згідно [14];
- додатки (якщо необхідно) (Додаток К).

Пояснювальна записка до курсової роботи повинна містити таку інформацію згідно обраного завдання:

Етап 1. Проведення аналізу.

Розглянути датчики та сенсори, описати відмінність між даними пристроями Визначити, що в Курсовій роботі будуть застосовані лише датчики. Провести аналіз існуючих датчиків. Проаналізувати певну кількість датчиків, зробити вибір в бік визначеного датчику, оскільки він має ряд переваг.

Розглянути існуючі платформи з обробки великих даних, вибрати платформу для роботи з над великими масивами даних, надати інформацію чому саме цю платформу.

Розглянути мови програмування, зробити вибір, які є найзручнішими для виконання задачі розробки системи моніторингу.

Етап 2. Написання скрипта (програми) щодо збору даних з датчиків.

Описати процес розробки програми й алгоритм, що дана програма повинна виконувати.

Етап 3. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків (Big Data).

Визначитися зі структурою даних та в якому вигляді ці дані повинні бути з імітовані, як потокові дані від датчиків, а також пояснити інформацію саме чому в такому вигляді потокові дані легше буде обробити за допомогою алгоритмів обробки надвеликих масивів даних.

Етап 4. Застосування методів обробки Big Data.

Розглянути та проаналізувати різні існуючі алгоритми, методи та платформи з обробки надвеликих масивів даних. Узгодити зі студентами, що відповідають за імітацію потокових даних, яку вибрати з проаналізованих платформу з певною мовою програмування.

5. Візуалізація даних.

Запропонувати, в якому вигляді зручніше представити оброблені дані, це стосується тільки візуалізаційної складової (графіки діаграми, голограми).

4.2. Вимоги до оформлення курсової роботи

Зміст розташовують безпосередньо після переліку скорочень та умовних позначень, починаючи з нової сторінки за номером 2.

Сторінки, на яких розміщено рисунки і таблиці, охоплюють загальною нумерацією сторінок роботи [13].

Для написання роботи використовують шрифт тексту – Times New Roman; розмір шрифту (кегль, вимірюється в пунктах) – 14 пт; міжрядковий інтервал – 1,5.

Текст роботи необхідно друкувати, залишаючи береги таких розмірів: ліве – 25 мм, праве – 15 мм, верхнє – 20 мм, нижнє – 20 мм.

Всі лінії, літери, цифри і знаки повинні бути однаково чорними впродовж усього документу.

Відступ у абзацах повинен бути однаковим впродовж усього тексту і дорівнювати 1,25 см.

При згадуванні в тексті прізвищ учених-дослідників ініціали, як правило, ставляться перед прізвищем (Х. М. Дейтел, а не Дейтел Х. М., як це прийнято в списку використаних джерел).

Зразок рисунка у тексті курсової роботи наведено на рис. 4.1.

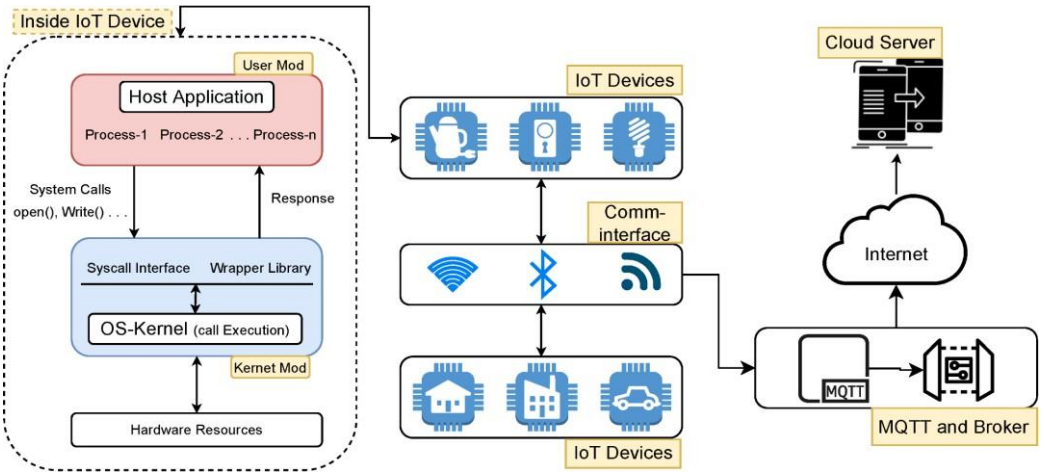

Рис. 4.1. Діаграма компонентів

Зразок формули у тексті курсової роботи виглядає наступним чином (рис.4.2.):

Сила підйому нагору F_g обчислюється за наступною формулою:

$$F_g = mgsin(\alpha) \qquad (2.6)$$

де

m – маса автомобіля, кг;

g – прискорення вільного падіння, м/с^2;

$sin(\alpha)$ – синус кута нахилу дороги.

Рис. 4.2. Оформлення формули

Зразок таблиці у тексті курсової роботи виглядає наступним чином:

Таблиця 4.1.

Опис існуючих команд в системі управління

Назва команди й вихідні параметри	Опис
stop_session	Зупиняє виконання системи імітації навантаження потокових даних.
gather	Збирає інформацію про існуючі пристрої. Повинна бути виконана першою, перед використанням команд які працюють з пристроями.
stop_device <device_id>	Зупиняє пристрій з вказаним ідентифікатором.
listen_device <device_id>	Переходить в режим прослуховування теми вказаного пристрою.

Посилання у тексті роботи

… у роботах [1-7] наведено …

… як зазначено у [4] …

… у розділі 4 …

… на рис. 1.3 …

… у табл. 3.2 … або (див. табл. 3.2)

… за формулою (3.1) …

… у рівняннях (1.7) – (1.9) …

… у додатку Б …

4.3. Вимоги до оформлення презентації роботи

Основними вимогами до презентації курсової роботи є: 1) структурованість - презентація має бути логічно та послідовно побудована, щоб її можна було легко зрозуміти та запам'ятати; 2) короткість - презентація повинна містити лише основну інформацію, щоб не втомлювати слухачів; 3) наочність; 4) відповідність до теми роботи.

Слайд 1. Назва роботи, ПІБ студента-автора курсової роботи та викладача (додатково надається вчений ступінь та звання).

Слайд 2. Мета та завдання роботи.

Слайд 3. Об’єкт та предмет дослідження.

Слайд 4. Методи дослідження.

Слайд 5. Зміст та структура роботи (коротко, основна частина та її розділи).

Слайд 6. Визначення основних понять, які досліджуються в роботі.

Слайди з результатами практичного дослідження. Навести найцікавіші приклади (оформлені курсивом, як і в тексті роботи), які найбільш яскраво демонструють результати.

Слайди із загальними висновками дослідження.

Останній слайд – Дякуємо за увагу!

У середньому кількість слайдів у презентації роботи – 12–14. Тобто презентація до захисту курсової роботи є своєрідними тезами до захисту роботи, Вашим планом і дозволеною опорою для виступу.

5. ПОРЯДОК ПОДАННЯ КУРСОВИХ РОБІТ ДО ЗАХИСТУ ТА ЗАХИСТ

5.1. Подання курсових робіт на перевірку

Курсова робота подається у двох примірниках: перший − у паперовій формі; другий – в електронному вигляді. Електронна версія курсової роботи має бути ідентичною паперовій. Разом з електронною версією роботи подається програмний продукт, який розроблений автором роботи, відповідно до завдання.

5.2. Перевірка на академічний плагіат

Відповідно до Положення про систему запобігання академічному плагіату в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» курсові роботи здобувачів вищої освіти підлягають перевірці на академічний плагіат, яка виконується на етапі допуску до захисту курсової роботи [15].

Перевірка на академічний плагіат курсових робіт здобувачів вищої освіти може здійснюється вибірково відповідно до рішення кафедри.

Текст остаточного варіанту курсового роботи подається здобувачем вищої освіти керівнику роботи в електронному вигляді на етапі допуску до захисту.

Відповідальна особа від кафедри здійснює процес перевірки роботи у Системі, яка генерує звіт подібності, та надає доступ керівнику до звіту подібності протягом трьох робочих днів після завантаження роботи.

Роздрукована частина звіту подібності, яка засвідчує відсоток збігів/ідентичності/схожості, завірена підписом відповідальної особи за перевірку робіт на плагіат, додається до курсового проекту/роботи.

Керівник здійснює експертну оцінку роботи з урахуванням звіту подібності, робить висновок про оригінальність роботи, на основі якого ухвалюється рішення щодо допуску роботи до захисту.

Допуск до захисту проводиться з врахуванням експертної оцінки керівника результатів щодо відсутності плагіату в роботі.

Якщо робота допускається до захисту, то здійснення заходів з перевірки на академічний плагіат вважається завершеним.

У випадку виявлення у роботі запозичень без належного оформлення посилань чи інших технічних недоліків в розділах, які не описують безпосередньо авторське дослідження, або якщо кількість цитат перевищує обсяг, виправданий поставленою метою роботи, робота повертається автору-здобувачу вищої освіти на доопрацювання з можливістю повторного подання на розгляд керівника. Повну копію звіту подібності (.pdf версію) здобувач вищої освіти може отримати на кафедрі за письмовим запитом.

У випадку незгоди з висновком про оригінальність роботи автор- здобувач вищої освіти має право подати апеляцію, яка буде розглянута у встановленому порядку Комісією з питань етики та академічної чемності Університету.

5.3. Процедура захисту курсових робіт

Захист курсових робіт відбувається у встановлений термін та присутності здобувача вищої освіти та комісії.

Процедура захисту включає:

- доповідь здобувача вищої освіти про зміст роботи;
- запитання до автора;
- відповіді здобувача вищої освіти на запитання членів комісії із захисту курсової роботи та осіб, присутніх на захисті;
- рішення комісії про оцінку роботи.

Залік проводиться у формі усного захисту курсової роботи перед комісією з проведення семестрового контролю

Доповідь здобувача вищої освіти не повинна перевищувати 5 хвилин. При цьому потрібно використовувати текст пояснювальної записки, разом з підготованою презентацією.

Комісія оцінює якість пояснювальної записки, текстового та графічного (ілюстративного) матеріалу: дотримання встановленого графіка виконання курсової роботи, сучасність та обґрунтування прийнятих рішень, правильність застосування методів аналізу і розрахунку, якість оформлення, виконання вимог нормативних документів, якість графічного матеріалу і дотримання вимог стандартів тощо.

Також оцінюється якість доповіді, ступінь володіння матеріалом, ступінь обґрунтування прийнятих рішень, вміння захищати свою думку, відповідей на запитання членів комісії з проведення семестрового контролю тощо.

Після захисту курсової роботи) комісія з проведення семестрового контролю підсумовує бали за стартовою складовою та складовою захисту, зводить до рейтингової оцінки та переводить до оцінок за університетською шкалою

5.4. Критерії оцінювання курсової роботи

У процесі визначення підготовчої складової оцінки за курсову роботу враховується низка важливих показників роботи [16]:

- актуальність дослідження;
- самостійність дослідження. Повнота проведеного аналізу;
- реалізація мети дослідження; відповідність висновків обраній темі;
- структура і логічність композиції роботи;
- обізнаність із класичними і новітніми науковими джерелами та володіння навичками реферативного опрацювання;

- відповідність обсягу роботи встановленим вимогам;

- відповідність вимогам наукового стилю; дотримання вимог щодо оформлення роботи;

- ступінь виконання завдань викладача та дотримання термінів виконання роботи.

Максимальна сума балів за допущену до захисту курсову роботу складає 40 балів.

Складова захисту курсової роботи.

Захист оцінюється за наступними критеріями [16]:

- структура і обсяг презентації роботи на захисті;
- мовна компетенція студента;
- повнота відповідей на поставлені запитання;
- вміння захищати власну думку;
- ступінь володіння матеріалом;
- культура захисту.

Максимальна сума балів за захист складає 60 балів.

Сума балів двох складових переводиться до оцінки згідно з таблицею 5.1:

Таблиця 5.1.

Відповідність рейтингових балів оцінкам

Бали	**Оцінка**
95–100	Відмінно
85–94	Дуже добре
75–84	Добре
65–74	Задовільно
60–64	Достатньо
Менше ніж 60	Незадовільно
Курсову роботу не допущено до захисту	Не допущено

6. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Положення про організацію освітнього процесу в КПІ ім. Ігоря Сікорського [Електронний ресурс]. - Режим доступу: http://osvita.kpi.ua/node/39

2. Освітньо-професійна програма «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» другого (магістерського) рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення [Електронний ресурс]. - Режим доступу: 121_OPPM_IPZIKFSE_2022.pdf

3. Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці. Курсової роботи. Навчальна програма (Силабус) [Електронний ресурс]. - Режим доступу: https://ipze.kpi.ua/silabusmag/

4. Історія Інтернету речей [Електронний ресурс]. - Режим доступу: https://perenio.ua/blog/the-history-of-the-internet-of-things.

5. Analysis of identifiers in IoT platforms. Digital Communications and Networks [Електронний ресурс]. - Режим доступу: https://www.sciencedirect.com/science/article/pii/S2352864818300671

6. AVSystem, “Top sensor types used in IoT” [Електронний ресурс]. - Режим доступу: https://www.avsystem.com/blog/iot-sensors-iot-actuators/

7. Ярослав Боцман Інтернет речей. ChannelForit Review №3. 2016 [Електронний ресурс]. - Режим доступу: https://www.snt.ua/storage/app/media/pdf/IoT.pdf

8. Allerin, «IoT data management: the benefits, challenges, and strategies» [Електронний ресурс]. - Режим доступу: https://www.allerin.com/blog/iot-data-management-the-benefits-challenges-and-strategies

9. Big Data, методи і техніки аналізу [Електронний ресурс]. - Режим доступу: https://www.it.ua/knowledge-base/technology-innovation/big-data-bolshie-dannye

10. Інтернет речей в енергетиці: у чому користь? [Електронний ресурс]. - Режим доступу: https://hub.kyivstar.ua/articles/internet-rechej-v-energeticzi-u-chomu-koristi

11. Smart metering: що це таке, навіщо та де застосовується [Електронний ресурс]. - Режим доступу: https://hub.kyivstar.ua/articles/smart-metering-shho-cze-take-navishho-ta-de-zastosovuyetsya

12. Пілотний проект з передачі даних Smart лічильників на основі стандарту NB-IoT [Електронний ресурс]. - Режим доступу: https://btk-center.com.ua/post/pilotniy-proekt-z-peredachi-danih-smart-lichilnikiv-na-osnovi-standartu-nb-iot-id18.html

13. ДСТУ 3008-2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – На заміну ДСТУ 3008–95; чинний з 2017–07–01. – Київ: ДП «УкрНДНЦ». 2016. – 26 с.

14. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання / Нац. стандарт України. – Вид. офіц. - [Уведено вперше; чинний від 2016-07-01]. – Київ: ДП «УкрНДНЦ». 2016. – 17 с.

15. Положення про систему запобігання академічному плагіату в КПІ ім. Ігоря Сікорського [Електронний ресурс]. - Режим доступу: https://osvita.kpi.ua/node/47

16. Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського [Електронний ресурс]. - Режим доступу: https://osvita.kpi.ua/node/37

Додаток А

Титульний лист курсової роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЕНЕРГЕТИЦІ

КУРСОВА РОБОТА

з дисципліни «Розробка застосунків Інтернету речей та сенсорних мереж в енергетиці»

на тему: **«Розробка системи моніторингу показників електролічильників. Імітація навантаження (потоків даних) – симуляція навантаження з датчиків (Big Data)»**

Студента(ки)___курсу___групи

_____
(прізвище, ініціали)

Керівник_____
(посада, вчене звання, науковий ступінь, ПІБ)

Національна шкала_____
Кількість балів:__оцінка: ETSI___

Члени комісії: _____ _____
(підпис) (прізвище, ініціали)

_____ _____
(підпис) (прізвище, ініціали)

_____ _____
(підпис) (прізвище, ініціали)

Київ – 202___

Додаток Б

Зразок оформлення змісту курсової роботи

Додаток В

Зразок оформлення переліку умовних позначень, символів і скорочень

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

а) україномовні

СКБД	Система керування базами даних
ЦОД	Центр оборбки даних

б) англомовні

API	Application programming interface
HTTP	HyperText Transfer Protocol
IoT	Internet of things
MQTT	Message Queuing Telemetry Transport
REST	Representational State Transfer
RGB	Red, Green, Blue

Додаток Г

Зміст вступу до курсової роботи

ВСТУП

Вступ – компонент наукової праці, в якому обґрунтовується необхідність проведення дослідження, розкривається ступінь розробки проблеми і значущість теми, підстави і вихідні дані для її дослідження.

Актуальність теми. Тут слід зазначити доцільність проведеного дослідження, мотивуючи існуючим (чи неіснуючим) рівнем сучасних досліджень, інтересом певних категорій споживачів (науки, виробництва, населення тощо).

Метою курсової роботи є … формулюється мета роботи

Для досягнення поставленої мети необхідно виконати наступні ***завдання***:

1) розглянути / ознайомитися з … ;

2) реалізувати / провести аналіз … ;

3) розробити … ;

4) дослідити … ;

5) створити … ;

6) експериментально перевірити

…

Відразу після цього, ставляться ***завдання*** в форматі «що потрібно зробити для досягнення мети?».

Написання завдань:

- пишуться списком: нумерованим чи ні;
- кожне завдання безпосередньо пов'язане з розділами/підрозділами;
- не використовувати повторювані слова. Наприклад, коли вже було слово «проаналізувати», в наступному пункті краще вибрати синонім;
- у висновках потрібно перераховувати результати за списком завдань.

Об'єктом дослідження є … формулюється об'єкт дослідження (це процес або явище, що породжує проблему й обране для дослідження)

Предметом дослідження виступає … формулюється предмет дослідження (деталізує конкретну проблему, його виділяють у межах об'єкта як складову частину).

Методи дослідження. Автор роботи має чітко окреслити всі використані у дослідженні методи для досягнення поставленої мети і завдань. Це має бути не звичайний номенклатурний перелік методів, а коротка й змістовна їх характеристика щодо конкретного застосування (що вивчалося за допомогою того чи того методу). Вибір методів повинен логічно випливати із змісту роботи, адекватно розкривати об'єкт і предмет дослідження.

Додаток Д

Зразок оформлення основної частини курсової роботи

3. РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ НАВАНТАЖЕННЯМ

Як було коротко згадано в розділі 2, система імітації потокового навантаження складається з двох практично незалежних частин: системи імітації даних, яка була розглянута в попередньому розділі та системи управління імітацією даних. Розподіл системи на дві незалежні програми дозволяє значно прискорити розробку, спросити тестування й відчиняє двері для інших розробників. Система управління навантаженням – це лише приклад, який супроводжується разом зі системою імітації навантаження й не є обов'язковим, оскільки управління може здійснюватися й лише за допомогою налаштування відповідних JSON файлів. Більше того, знаючи стандарт, за яким спілкуються між собою ці дві системи, будь-хто може розробити власну, більш кращу, систему управління, оснащену, наприклад, графічним інтерфейсом й можливістю візуалізації.

3.1. Загальний огляд системи управління навантаженням

На рисунку 3.1 схематично представлено місце системи управління в загальній архітектурі. Ця система отримала назву console, оскільки фактично реалізує консоль, за допомогою якої, з використанням набору команд, здійснюється керування навантаженням.

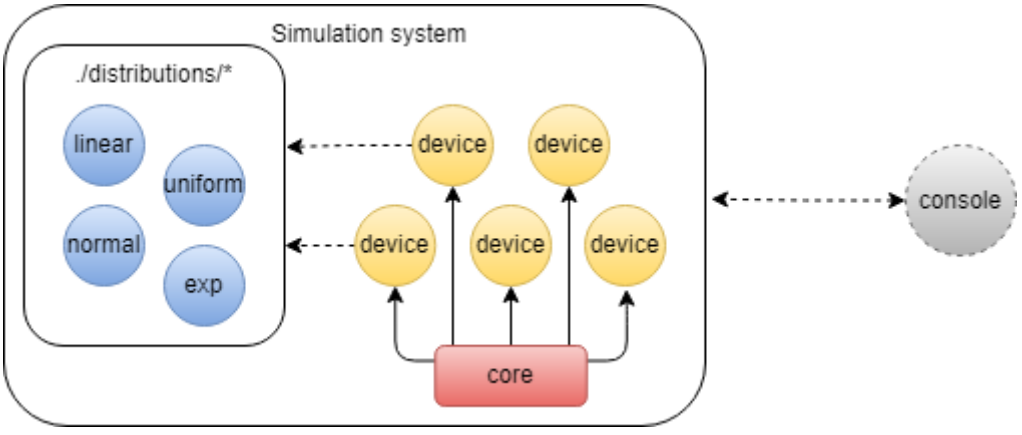

Рисунок 3.1. Місце системи управління в загальній архітектурі

Система управління реалізована як окремий Python файл. Для забезпечення роботи цієї системи, необхідно виконання двох умов:

1. Користувач повинен задати конфігураційний файл з використанням нотації json й вказати інформацію брокера відповідну тій, яка використовується системою імітації навантаження.

2. Перед запуском системи, система імітації навантаження вже повинна бути запущена й ідентифікатор UUID сесії якої повинен бути заданий як вихідний аргумент для системи управління – за допомогою цього ідентифікатора буде здійснюватися спілкування між програмами.

Спілкування відбувається з використанням протоколу MQTT з використанням двох тем:

1. dfsm:UUID – За цією темою відбувається надсилання команд зі сторони системи управління. Усі пристрої, для яких не було вимкнено режим керування, в момент з'єднання підписуються на цю тему й прослуховують її до кінця своєї роботи.

2. dfsm:UUID:response – За цією темою відбувається надсилання відповідей зі сторони пристроїв назад до системи управління. Важливо зауважити, що цю тему можуть прослуховувати усі клієнти брокера. В разі якщо це неприйнятно й користувач хоче уникнути втрати приватності, додатково може бути використана опція шифрування даних з використанням асиметричного ключа, за допомогою бібліотеки rsa [10] – в цьому випадку необхідно також отримати доступ до приватного ключа, який було створено в момент запуску системи імітації навантаження.

Використання декількох консолей також можливо, хоча при цьому система імітації навантаження не буде розрізняти консолі між собою й надсилати повідомлення до обох одночасно. Тим не менш, це може бути корисно, наприклад, коли необхідно прослуховувати декілька пристроїв одночасно й відображати результати на екран або у файли.

3.2. Стани системи управління

Систему було розроблено з використанням скінченних автоматів. Кожен стан виконує дві функції:

1. Кожного логічного фрейму виконує певну дії, наприклад, створення запиту на введення певної команди.

2. Оброблює вихідні пакети від системи імітації навантаження.

На початку, система знаходиться в стані Menu, який відповідальний за виведення загальної інформації про систему й зчитування наступної команди. Кожна команда має свій власний стан, проте оскільки більшість команди є достатньо простими, програмно вони були реалізовані лише як окремі методи стану Menu. Якщо клієнт робить запит на початок прослуховування певного пристрою, система переходить в стан Listening, який підписується на тему, на яку публікує повідомлення вказаний пристрій й виводить усі вхідні повідомлення зі вказаної теми. Також Listening прослуховує клавіатуру користувача й повертається до стану Menu в разі генерування сигналу SIGINT (ctrl-c). Поточний вигляд скінченного автомата представлено на рисунку 3.2.

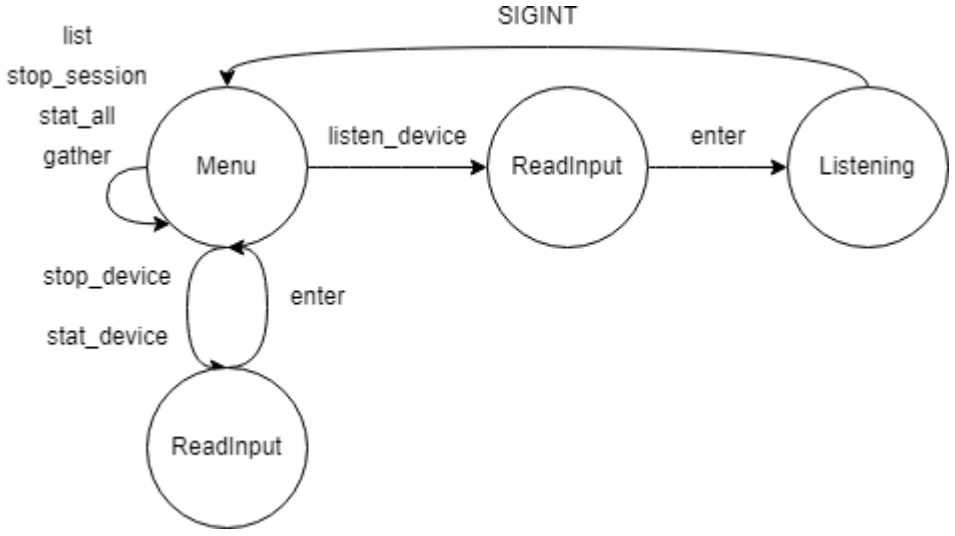

Рисунок 3.2. Поточний вигляд скінченного автомата системи управління

3.3. Існуючі команди системи управління

На даний момент, система підтримує 7 команд, які представлені в таблиці 3.1. Потенційно також можуть бути додані такі команди, як deactive/activate для тимчасового вимкнення та увімкнення штучного пристрою, modify для зміни параметрів пристрою в реальному часі, plot для схематичного відображення згенерованих даних. Як вже згадувалось раніше, користувачі також можуть розробити власну систему управління або розширити файл dfsm_console.py й додати власні команди. Завдяки використання нотації JSON, можна легко побудувати застосунки, які дозволять спросити заповнення й збирання певної інформації: наприклад, дозволять створювати бажані розподіли за допомогою графічних програм.

Таблиця 3.1.

Опис існуючих команд в системі управління

Назва команди й вихідні параметри	Опис
stop_session	Зупиняє виконання системи імітації навантаження потокових даних.
gather	Збирає інформацію про існуючі пристрої. Повинна бути виконана першою, перед використанням команд які працюють з пристроями.
stop_device <device_id>	Зупиняє пристрій з вказаним ідентифікатором.
listen_device <device_id>	Переходить в режим прослуховування теми вказаного пристрою.
list	Виводить загальну інформацію про існуючі пристрої.

Продовження таблиці 3.1.

stat_all	Виводить інформацію по всім існуючим пристроям.
stat_device <device_id>	Виводить статистичну інформацію вказаного пристрою. В разі, якщо пристрій не збирає статистику або gather не було викликано – статистика не буде виведена.

Висновки до розділу 3

Система управління є допоміжною програмою, яка дозволяє користувачам керувати імітацією навантаження в реальному часі, а також збирати статистичну інформацію й прослуховувати обрані штучні пристрої. Ця система є прикладом реалізації програми для взаємодії зі системою навантаження й не є обов'язковою для використання. Керуючись принципами, за якими працює система управління, користувач може створити власну систему управління або вдосконалити існуючу.

Додаток Е

Зміст розділу висновків до курсової роботи

ВИСНОВКИ

У курсовій роботі … описати, що зроблено (Висновки містять викладення найважливіших наукових та практичних результатів, одержаних у результаті проведеного дослідження. Пункти висновків мають бути лаконічними, змістовними, вони повинні читатися і сприйматися легко й однозначно).

Основні результати проведеної роботи полягають у наступному.

1. Наведено … .
2. Розглянуто … .
3. Реалізовано … .
4. Розроблено … .
5. Запропоновано … .
6. Досліджено … .
7. Здійснено … .

Додаток Ж

Оформлення списку використаних джерел

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – На заміну ДСТУ 3008–95; чинний з 2017–07–01. – Київ: ДП «УкрНДНЦ». 2016. – 26 с.

2. Історія Інтернету речей [Електронний ресурс]. - Режим доступу: https://perenio.ua/blog/the-history-of-the-internet-of-things.

3. Barnes-Hut Galaxy Simulator. Курс 126 Комп’ютерних наук Прінстонського університету [Електронний ресурс]. – Режим доступу: http://introcs.cs.princeton.edu/java/assignments/barnes-hut.

Додаток К
Оформлення додатку із програмним кодом

Додаток А
ЛІСТИНГ ПРОГРАМИ

dfsm_core.py

```
import sys
import logging
import json
import signal
import uuid
from threading import Thread

import dfsm_settings as settings
from dfsm_device import *

registered_devices = {}

class DiscreteSampler:
    def __init__(self, data):
        points_sum = sum(data['points'])
        self.rng = data['range']
        self.points = [x / points_sum for x in data['points']]
        self.acc_points = [x for x in self.points]
        self.acc_points[0] = self.points[0]
        for i in range(1, len(self.acc_points)):
            self.acc_points[i] = self.acc_points[i - 1] + self.points[i]

    def inv_cdf(self, params):
        e = params['e']
        bucket_idx = 0
        prev_value = 0
        for i, value in enumerate(self.acc_points):
            if e >= prev_value and e < value:
                bucket_idx = i
                break
            prev_value = value

        bucket_size = 1.0 / len(self.points)

        if bucket_idx == 0:
            # extrapolate to 0
            x = (e / self.acc_points[0]) * bucket_size
        else:
            # interpolate between bucket's edge acc points
            e1 = self.acc_points[bucket_idx - 1]
            e2 = self.acc_points[bucket_idx]
            s = (e - e1) / (e2 - e1) * bucket_size
            x = bucket_idx * bucket_size + s

        rng = self.rng
        return x * (rng[1] - rng[0]) + rng[0]

def parse_input(argv):
    return {}

def load_json(fname):
    opened_file = open(fname)
```