

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Марченко Вадим Генрихович

(ПІБ)

академічної групи

122-19-3

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

Розробка комп'ютерної системи контролю вуличного

освітлення

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтингов ою	інституцій ною	
кваліфікаційної роботи	<i>проф. Лактіонов І.С.</i>			
розділів:				
спеціальний	<i>проф. Лактіонов І.С.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу
бакалавра
(назва освітньо-кваліфікаційного рівня)

студента 122-19-3 Марченко В.Г.
(група) (прізвище та ініціали)
тема кваліфікаційної роботи Розробка комп'ютерної системи
контролю вуличного освітлення

затверджена наказом ректора НТУ «ДП» від 16.05.2023 № 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2023 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2023 р.

Завдання видав _____ проф. Лактіонов І.С.
(підпис) (посада, прізвище, ініціали)
Завдання прийняв до виконання _____ Марченко В.Г.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: 63 с., 12 рис., 3 дод., 20 джерел.

Об'єкт розробки: комп'ютерна система контролю вуличного освітлення.

Мета кваліфікаційної роботи: підвищення ефективності програмно-технічних засобів керування вуличним освітленням за рахунок створення комп'ютерної моделі забезпечення системи контролю вуличного освітлення.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробку програми, описано роботу програми, алгоритми і структуру її функціонування, а також виклик та завантаження програми, визначено вхідні й вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розроблення програмного забезпечення, проведений підрахунок вартості робіт по створенню програми та розраховано час на його створення.

Практичне значення полягає в створенні комп'ютерної моделі системи в середовищі CupCarbon із використанням аналогових та цифрових давачів, приймачів та передавачів мережевих повідомлень на основі технологій LoRa і Zigbee.

Список ключових слів: РОЗУМНЕ МІСТО, ПРИЙМАЧ, ПЕРЕДАВАЧ, РОУТЕР, СКРІПТ, СЕНСОРНИЙ ВУЗОЛ, СИСТЕМА ОСВІТЛЕННЯ.

ABSTRACT

Explanatory note: 63 pp., 12 fig., 3 appendix, 20 sources.

Object of development: computer system of control of street lighting.

The purpose of the qualification work: creation of software computer system control of street lighting. The system transmits lighting data to the server

In the introduction, the analysis and current state of the problem is considered, the purpose of the qualification work and the field of its application are specified, the justification of the relevance of the topic is given, and the statement of the task is clarified.

In the first section, the subject area is analyzed, the relevance of the task and the purpose of the development is determined, the task statement is formulated, and the requirements for software implementation, technologies and software tools are specified.

In the second section, available solutions are analyzed, platforms for development are selected, program design and development has been performed, program operation, algorithm and structure of its functioning are described, as well as program calling and loading, input and output data are determined, and the composition of technical means parameters is characterized.

In the economic section, the labor-intensiveness of software development is defined, the cost of robots for creating the program is calculated, and the time for its creation is calculated.

The practical value is to create a computer model of the system in the CupCarbon environment, with the use, adjustment and restoration of operational efficiency of analog and digital sensors, receivers and transmitters, LoRa and Zigbee.

Keywords: SMART CITY, RECEIVER, TRANSMITTER, ROUTER, SCRIPT, SENSOR NODE, LIGHTING SYSTEM.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
ВСТУП.....	Ошибка! Закладка не определена.
РОЗДІЛ 1	8
1.1 Загальні відомості з предметної галузі:	8
1.2 Призначення розробки та галузь застосування:.....	14
1.3 Підстава для розробки	17
1.4 Постановка завдання.....	17
1.5 Підрозділ «вимоги до програми або програмного виробу»	17
1.5.1 Вимоги до функціональних характеристик.....	17
1.5.2 Вимоги до інформаційної безпеки.	17
1.5.3 Вимоги до складу та параметрів технічних засобів.	18
1.5.4 Вимоги до інформаційної та програмної сумісності.....	18
РОЗДІЛ 2	19
2.1 Функціональне призначення системи.....	19
2.2 Опис застосованих математичних методів.....	19
2.3 Опис використаних технологій та мов програмування.....	19
2.4 Опис структури системи та алгоритмів її функціонування.....	20
2.5 Обґрунтування та організація вхідних та вихідних даних програми.....	22
2.6 Опис розробленої системи.	23
2.6.1 Використані технічні засоби.....	23
2.6.2 Використані програмні засоби.....	25
2.6.3 Виклик та завантаження програми.....	37
2.6.4 Опис інтерфейсу користувача.....	40
ЕКОНОМІЧНИЙ РОЗДІЛ	42
3.1 Розрахунок трудомісткості та вартості розробки програмного продукту	42
3.2. Розрахунок витрат на створення програми	44
Висновок.	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТОК Б	59
ДОДАТОК В	60

ВСТУП

Комп'ютерна система контролю вуличного освітлення потрібна для безпечного пересування мешканців будинку, забезпечення руху автотранспорту, декоративного підсвічування будівель або їх окремих елементів, освітлення реклами на білбордах.

Увімкнення та вимкнення вуличних світильників здійснюється в декількох способів керування світлом: ручне, дистанційне та автоматичне. При Увімкненні та вимкненні вуличного освітлення в ручному режимі: кожне джерело світла або їхня група управляється оператором безпосередньо на місці. Цей спосіб найдавніший.

З часом технології розвивалися – замість операторів керувати освітленням стали службовці енергорозподільчих мереж. Робили працівники служб це дистанційно, включаючи чи вимикаючи рубильник. Внаслідок дій напруга подається в мережу або, навпаки, припиняється.

Управління за допомогою автоматики – найбільш просунутий спосіб керування світлом. Вмикання та вимикання світла здійснюється за рахунок використання датчиків, що діють за певним алгоритмом. У результаті система висвітлення працює без безпосередньої участі людини.

Існує дві обставини, що диктують перехід на автоматичне керування:

– Найчастіше будувати окремі підстанції для вуличного освітлення економічні не вигідно. Нинішні трансформатори перетворюють напругу всім споживачів електрики на заданій території.

– Для централізованого контролю за включенням та відключенням світильників знадобилося б підтягувати до кожної підстанції окремий кабель, що тільки підвищить і так великі витрати.

У зв'язку з цим розпочався масовий перехід на автоматичні системи. На початку розвитку технології принцип управління був простий: на підстанціях монтувалися прилади, що контактували з датчиками освітленості.

Тому проблема, розглянута в даній кваліфікаційній роботі, є актуальною та має широке практичне значення.

Метою даної роботи є розробка системи освітлення при змінюванні погоди та інтеграція її в Smart City.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

«Термін «інтернет речей», мабуть, зобов'язаний своєю появою Кевіну Ештону, який у 1997 р., працюючи на компанію Proctor and Gamble, для управління системою постачання застосував технологію радіочастотної ідентифікації (RFID). Завдяки цій роботі в 1999 р. його запросили до Массачусетського технологічного інституту, де він з групою однодумців організував дослідницький консорціум Auto-ID Center (докладнішу інформацію можна знайти на сайті www.smithsonianmag.com/innovation/kevinashton-describes-the-internet-of-things-180953749/). З того часу Інтернет речей здійснив перехід від простих радіочастотних міток до екосистеми та індустрії, яка до 2020 р. залучить, створить або поглине 5 трлн доларів зі 100 трлн світового ВВП, тобто 6% світового ВВП. Аж до 2012 р. ідея підключення речей до інтернету переважно ставилася до смартфонів, планшетів, ПК та ноутбуків. По суті, до тих речей, які у всіх відносинах виступають як комп'ютер. До цього, з появи перших боязких зачатків інтернету (таких як створена 1969 р. мережу ARPANET), більшості технологій, у яких будується інтернет речей, просто існувало. До 2000 р. більшість пристроїв, які можна було підключити до інтернету, були комп'ютерами різних розмірів» [1].

Що таке smart city; його принципи та приклади [2].

«Smart City (далі SM) використовують різноманітне програмне забезпечення, користувацькі інтерфейси та комунікаційні мережі поряд з Інтернетом речей (IoT), щоб надавати підключені рішення для населення. З них IoT є найважливішим. IoT – це мережа підключених пристроїв, які спілкуються та обмінюються даними. Це може включати все, від транспортних засобів до побутової техніки та вуличних датчиків. Дані, зібрані з цих пристроїв, зберігаються в хмарі або на серверах, щоб дозволити покращити ефективність як

державного, так і приватного секторів і забезпечити економічні вигоди та покращення життя громадян.

Поряд з рішеннями IoT розумні міста також використовують такі технології, як:

Інтерфейси прикладного програмування (API);

Штучний інтелект (AI);

Послуги хмарних обчислень;

Приладові панелі;

Машинне навчання;

Комунікації між машинами;

Mesh мережі.

Поєднання автоматизації, машинного навчання та IoT дозволяє використовувати технології розумного міста для різноманітних застосувань. Наприклад, розумне паркування може допомогти водіям знайти місце для паркування, а також дозволить здійснювати цифрові платежі.

Іншим прикладом може бути інтелектуальне управління дорожнім рухом для моніторингу транспортних потоків і оптимізації світлофорів для зменшення заторів, а послугами спільного використання поїздок також можна керувати за допомогою розумної міської інфраструктури.

Функції «розумного міста» також можуть включати енергозбереження та екологічну ефективність, наприклад вуличні ліхтарі, які тьмяніють, коли дороги порожні. Такі технології інтелектуальної мережі можуть покращити все: від експлуатації до технічного обслуговування та планування до електропостачання.

Ініціативи «розумного міста» також можна використовувати для боротьби зі зміною клімату та забрудненням повітря, а також для управління відходами та санітарії за допомогою підключених до Інтернету систем збору сміття, урн і систем управління автопарком.

Крім послуг, розумні міста дозволяють забезпечити заходи безпеки, такі як моніторинг районів з високим рівнем злочинності або використання датчиків для раннього попередження про такі події, як повені, зсуви, урагани або посухи.

Розумні будівлі також можуть запропонувати керування простором у режимі реального часу або моніторинг стану конструкції та зворотний зв'язок, щоб визначити, коли потрібен ремонт. Громадяни також можуть отримати доступ до цієї системи, щоб повідомити чиновників про будь-які проблеми, такі як вибоїни, а датчики також можуть контролювати проблеми інфраструктури, такі як витoki у водопровідних трубах.

Крім того, технологія розумного міста може підвищити ефективність виробництва, міського сільського господарства, використання енергії тощо.

Розумні міста можуть об'єднувати різноманітні послуги, щоб надавати об'єднані рішення для громадян.

Концепція розумних міст виникла ще в 1960-х і 1970-х роках, коли Бюро аналізу громадськості США почало використовувати бази даних, аерофотознімки та кластерний аналіз для збору даних, спрямування ресурсів і випуску звітів з метою спрямування послуг, пом'якшення наслідків стихійних лих і зменшення бідності. Це призвело до створення першого покоління розумних міст.

Перше покоління розумного міста було створено постачальниками технологій, щоб зрозуміти вплив технологій на повсякденне життя. Це призвело до другого покоління розумного міста, яке розглядало, як розумні технології та інші інновації можуть створити об'єднані муніципальні рішення. Третє покоління розумного міста відібрало контроль у постачальників технологій і керівників міст, натомість створивши модель, яка залучає громадськість і забезпечує соціальну інтеграцію та залучення громади.

Ця модель третього покоління була прийнята Віднем, який створив партнерство з місцевою компанією Wien Energy, що дозволило громадянам інвестувати в місцеві сонячні електростанції, а також співпрацювати з громадськістю для вирішення проблем гендерної рівності та доступного житла. Таке прийняття продовжується в усьому світі, зокрема у Ванкувері, де 30 000 громадян спільно створили План дій Ванкувера «Найзеленіше місто 2020».

Розумні міста виконують чотири кроки, щоб покращити якість життя та забезпечити економічне зростання за допомогою мережі підключених пристроїв Інтернету речей та інших технологій. Ці кроки такі:

Збір – розумні датчики збирають дані в реальному часі.

Аналіз – дані аналізуються, щоб отримати уявлення про роботу міських служб і операцій.

Комунікація – результати аналізу даних повідомляються особам, які приймають рішення.

Дія – вжито заходів для покращення роботи, управління активами та покращення якості життя мешканців міста.

ІКТ-структура об'єднує дані в реальному часі з пов'язаних активів, об'єктів і машин для покращення прийняття рішень. Однак, крім того, громадяни можуть залучатися та взаємодіяти з екосистемами розумного міста за допомогою мобільних пристроїв і підключених транспортних засобів і будівель. Поєднуючи пристрої з даними та міською інфраструктурою, можна скоротити витрати, підвищити стійкість і оптимізувати такі фактори, як розподіл енергії та збір сміття, а також запропонувати зменшення заторів на дорогах і покращити якість повітря.

Чому SM важливо. 54 % населення світу живе в містах, і очікується, що до 2050 року цей показник зросте до 66 %, додавши ще 2,5 мільярда людей до міського населення протягом наступних трьох десятиліть. З таким очікуванням зростанням населення виникає потреба в управлінні екологічною, соціальною та економічною стійкістю ресурсів.

Розумні міста дозволяють громадянам і органам місцевого самоврядування працювати разом, щоб запускати ініціативи та використовувати розумні технології для управління активами та ресурсами у зростаючому міському середовищі.

Навіщо нам потрібен SM. Розумне місто має забезпечувати міське середовище, яке забезпечує високу якість життя жителів, а також сприяє

економічному зростанню. Це означає надання набору об'єднаних послуг громадянам зі зниженими витратами на інфраструктуру.

Це стає все більш важливим у світлі майбутнього зростання населення в міських районах, де буде потрібно більш ефективного використання інфраструктури та активів. Сервіси та програми «розумного міста» уможливають ці покращення, що призведе до вищої якості життя громадян.

Удосконалення «розумного міста» також забезпечує нову цінність існуючої інфраструктури, водночас створюючи нові потоки доходів і ефективність роботи, щоб допомогти заощадити гроші як урядам, так і громадянам.

SM Проблеми. Попри всі переваги «розумних» міст, є й труднощі, які необхідно подолати. До них належать урядовці, які дозволяють широку участь громадян. Також існує потреба в тому, щоб приватний і державний сектори об'єдналися з мешканцями, щоб кожен міг зробити позитивний внесок у громаду.

Проекти розумних міст мають бути прозорими та доступними для громадян через портал відкритих даних або мобільний додаток. Це дозволяє мешканцям працювати з даними та виконувати особисті завдання, такі як оплата рахунків, пошук ефективних варіантів транспорту та оцінка споживання енергії в будинку.

Усе це вимагає надійної та безпечної системи збору та зберігання даних, щоб запобігти злому чи зловживанню. Дані розумного міста також мають бути анонімними, щоб запобігти виникненню проблем із конфіденційністю.

Найбільшою проблемою, ймовірно, є підключення, коли тисячі або навіть мільйони пристроїв Інтернету речей повинні підключатися та працювати в унісон. Це дозволить об'єднати послуги та вносити постійні покращення в міру зростання попиту.

Крім технологій, розумні міста також повинні враховувати соціальні фактори, які створюють культурну тканину, привабливу для мешканців і

створюють відчуття місця. Це особливо важливо для тих міст, які створюються з нуля і потребують залучення жителів.

SM Безпека. Розумні міста пропонують багато переваг для підвищення безпеки громадян, таких як підключені системи спостереження, інтелектуальні дороги та моніторинг громадської безпеки, але як щодо захисту самих розумних міст?

Необхідно забезпечити захист «розумних» міст від кібератак, злому та крадіжки даних, а також забезпечити точність даних, які надсилаються.

Щоб керувати безпекою розумних міст, необхідно впроваджувати такі заходи, як фізичні сховища даних, стійке керування автентифікацією та ідентифікаційні рішення. Громадяни мають довіряти безпеці розумних міст, а це означає, що уряд, підприємства приватного сектору, розробники програмного забезпечення, виробники пристроїв, постачальники енергії та менеджери мережевих послуг повинні працювати разом, щоб надавати інтегровані рішення з основними цілями безпеки. Ці основні цілі безпеки можна розбити таким чином:

Доступність – дані мають бути доступними в режимі реального часу з надійним доступом, щоб переконатися, що вони виконують свою функцію моніторингу різних частин інфраструктури розумного міста.

Нечесність – дані мають бути не лише доступними, але й точними. Це також означає захист від маніпуляцій ззовні

Конфіденційність – Конфіденційні дані повинні бути конфіденційними та захищеними від несанкціонованого доступу. Це може означати використання брандмауерів або анонімізацію даних

Підзвітність – користувачі системи повинні нести відповідальність за свої дії та взаємодію з конфіденційними системами даних. У журналах користувачів має бути записано, хто має доступ до інформації, щоб забезпечити підзвітність у разі виникнення проблем

Законодавство вже впроваджується в різних країнах, наприклад Закон про покращення кібербезпеки Інтернету речей у Сполучених Штатах, щоб

допомогти визначити та встановити мінімальні вимоги безпеки для підключених пристроїв у розумних містах.

Міста в усьому світі перебувають на різних етапах розвитку та впровадження розумних технологій. Однак є кілька тих, хто йде попереду, ведучи шлях до створення повністю розумних міст. До них належать:

London, England.

Toronto, Canada.

Hong Kong, China.

Barcelona, Spain.

Melbourne, Australia.

Створення інтелектуальних підключених систем для наших міських територій забезпечує багато переваг для громадян у всьому світі не лише для покращення якості життя, але й для забезпечення стійкості та найкращого використання ресурсів.

Ці рішення залежать від єдиного підходу з боку уряду, а також приватного сектору та самих мешканців. Однак за наявності належної підтримки та інфраструктури розумні міста можуть використовувати такі досягнення, як Інтернет речей, щоб покращити життя мешканців і створити рішення для спільного життя для зростаючого глобального міського населення.

TWI має досвід у багатьох базових технологіях нового розумного майбутнього, включаючи датчики, рішення для електрифікації та виробництва електроенергії» [3].

1.2. Призначення розробки та галузь застосування

«Правильне освітлення має забезпечувати рівномірний розподіл світлового потоку. Навіть чітко спрямоване світло не повинно мати окремих променів. При попаданні такого освітлення на сітківку ока можна отримати травму, тимчасове погіршення зору, больові відчуття» [4].

«Існуючи види сервісів для освітлення при зміні погоди, як його встановити та чому він працює

Розумні ліхтарі можна запрограмувати на ряд дій, наприклад, вимикати, коли ви заводите автомобіль, або вмикати, коли Fitbit виявляє, що ви встали з ліжка.

Щоб налаштувати цю автоматизацію, вам потрібно лише аккаунт IFTTT і сумісна розумна лампочка зі зміною кольору, наприклад зразки закордонні фірми Lix, Philips Hue або LED.

«IFTTT походить від умовного оператора програмування «якщо це, то це». Компанія надає програмну платформу, яка об'єднує додатки, пристрої та служби від різних розробників, щоб запуснути одну або кілька автоматизацій із залученням цих додатків, пристроїв і служб [5].»

Це допоможе продовжити та активувати відповідний канал для лампочки, а також канал погоди на IFTTT. Це можна зробити, перейшовши на IFTTT.com і натиснувши Канали у верхньому правому куті. Знайдіть і виберіть марку лампочки, увійдіть у свій обліковий запис і авторизуйте з'єднання IFTTT.

Щоб активувати канал погоди, знайдіть його у списку каналів, виберіть його, натисніть «Підключити», знайдіть своє поточне місто та натисніть «Підключити» ще раз.

Почніть із натискання «Мої налаштування» у верхній частині домашньої сторінки IFTTT.com.

- Натисніть Create a Recipe у правій частині сторінки.
- Для тригерного каналу («ця» частина рецепту) виберіть «Погода».
- Є маса варіантів для тригера або погодних умов, які спричинять запуск рецепта. Для цього прикладу оберіть Поточний стан зміни на як тригер.
- Натисніть спадне меню та виберіть Дощ. Натисніть «Створити тригер», щоб перейти.
- Знайдіть і виберіть відповідний канал дії для вашої розумної лампочки - у цьому випадку LIFX. Виберіть «Змінити колір світла» як дію та натисніть «Створити дію», щоб перейти.

- Виберіть, які вогні ви хочете змінити, коли почнеться дощ. Виберіть колір, який представлятиме дощ. (Зелений, жовтий, червоний і рожевий, як правило, використовуються для відображення на радарі інтенсивності дощу, а різні тони синього використовуються для позначення снігу, тому ви можете зарезервувати синій, якщо плануєте створити подібний рецепт для снігу). Виберіть рівень яскравості, чи бажаєте ви, щоб лампочка вмикалася, коли запускається рецепт, і тривалість переходу (чи бажаєте ви, щоб зміна відбувалася швидко чи повільно).

- Натисніть «Створити дію», щоб завершити налаштування» [6].

«IFTTT походить від умовного оператора програмування «якщо це, то це». Компанія надає програмну платформу, яка об'єднує додатки, пристрої та служби від різних розробників, щоб запуснути одну або кілька автоматизацій із залученням цих додатків, пристроїв і служб.

Автоматизація здійснюється за допомогою аплетів, які схожі на макроси, які з'єднують кілька програм для виконання автоматизованих завдань. Ви можете ввімкнути або вимкнути аплет за допомогою веб-сайту або мобільних програм IFTTT (та / або віджетів IFTTT мобільних програм). Ви також можете створювати власні аплети або створювати варіанти існуючих за допомогою зручного, простого інтерфейсу IFTTT.

Такі різноманітні розробники, як Ring і BMW, сплачують IFTTT щорічну плату за надання аплетів на платформі IFTTT. Партнерство з британським стартап-банком Monzo дозволяє клієнтам автоматично знімати кошти з накопичувача «на чорний день», коли йде дощ, або «винагороджувати» себе кожного разу, коли вони йдуть у спортзал.

Підтримка JavaScript допомагає партнерам IFTTT створювати надійні аплети порівняно з більш обмеженими рецептами IFTTT минулих років, сказав Тіббетс. Ви можете створити власний JavaScript, який автоматично фільтруватиме речі, щоб аплет, наприклад, увімкнув декілька світильників у вашому домі, якщо ви приїдете після 6 вечора. або просто світло під'їзду, якщо

ви приїдете додому до 6 вечора, сказав Тіббетс. Така функціональність неможлива з простішими рецептами, але здійснима з аплетами. » [7]

1.3. Підстава для розробки

Підставою для розробки кваліфікаційної роботи на тему “Комп’ютерна система контролю вуличного освітлення” є наказ по Національному технічному університету «Дніпровська політехніка» 350-с від 16.05.2023.

1.4. Постановка завдання

Мета кваліфікаційної роботи: підвищення ефективності програмно-технічних засобів керування вуличним освітленням за рахунок створення комп’ютерної моделі забезпечення системи контролю вуличного освітлення.

1.5. Підрозділ «вимоги до програми або програмного виробу»

1.5.1. Вимоги до функціональних характеристик

Система складається з декількох маршрутизаторів які отримують дані з аналогових датчиків, при зміні значення у датчика, маршрутизатор відсилає повідомлення з своїм ім’ям, ідентифікатором та значенням на сервер

1.5.2. Вимоги до інформаційної безпеки.

Певних вимог до інформаційної безпеки не висувається, оскільки розповсюдження програми в широкому загалі не планується. Проте, в разі необхідності, доступу до сторінок програми для кваліфікованих фахівців, потрібно передбачити можливість введення паролю, доступ до виклику якого не має бути явним.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для повноцінного функціонування програмного забезпечення необхідні:

1. персональний комп'ютер
2. клавіатура;
3. миша.

Певних вимог до потужності комп'ютера не висувається, оскільки достатньо запуску на ньому операційної системи.

1.5.4 Вимоги до інформаційної та програмної сумісності.

Розроблюване програмне забезпечення сумісне з операційними системами Windows XP, 7, 8 чи 10 32 чи 64-розрядних версій з JDK 8.0 та CupCarbon.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення системи

«Вуличні ліхтарі допомагають забезпечити безпеку нічним працівникам, додають краси та яскравості нічному вигляду місцевості та багато інших корисних причин.

Вуличне освітлення має важливе значення, починаючи від штучного збільшення тривалості світла і закінчуючи забезпеченням безпеки; мати безпечне водіння вночі [8].»

Застосування розроблюваної моделі має скоротити затрати на реалізацію моделі та подальші випробування на розробленій моделі будуть менш затратні ніж на натурному об'єкті

Безпосередньо після завантаження програми `cupcarbon` та моделі користувач повинен перейти до командного рядка та за допомогою команди `cd` перейти до теки знаходження програми `cupcarbon` (`cd C:\Users\Achem\Desktop\cc`) після треба виконати команду `java -jar cupcarbon.jar` Після запуску програми, користувач обирає у меню зверху `Project->Open Project` та вибирає файл у текі проекту з розширенням `.cup`. Далі щоб запустити проект `Simulation->Run Simulation`.

2.2. Опис застосованих математичних методів

При розробленні програмного для кваліфікаційної роботи не застосовуються математичні методи

2.3. Опис використаних технологій та мов програмування

При розробленні даної кваліфікаційної роботи застосовується Sensript. Це скрипт, який використовується для програмування сенсорних вузлів симулятора CupCarbon. Це сценарій, у якому змінні не оголошено і немає типів, але їх можна ініціалізувати (команда set). Можна використовувати функцію інструкції для додавання складних та додаткових функцій, запрограмованих на Java (in a source code mode only).

2.4 Опис структури системи та алгоритмів її функціонування.

Програмне забезпечення для кваліфікаційної роботи розроблено за допомогою скрипта Sensript у середовищі розробки CupCarbon.

«CupCarbon — це симулятор Smart City та бездротової сенсорної мережі Інтернету речей (SCI-WSN). Його метою є розробка, візуалізація, налагодження та перевірка розподілених алгоритмів для моніторингу, збору даних про навколишнє середовище тощо, а також створення сценаріїв навколишнього середовища, таких як пожежі, газ, мобільні пристрої, і загалом у рамках освітніх і наукових проектів. Це не тільки може допомогти наочно пояснити основні поняття сенсорних мереж і принцип їх роботи; він також може допомогти вченим перевірити їхні бездротові топології, протоколи тощо» [9].

В загальному вигляді, робота з реєстратором розрахункових операцій виглядає наступним чином (рис. 2.1).

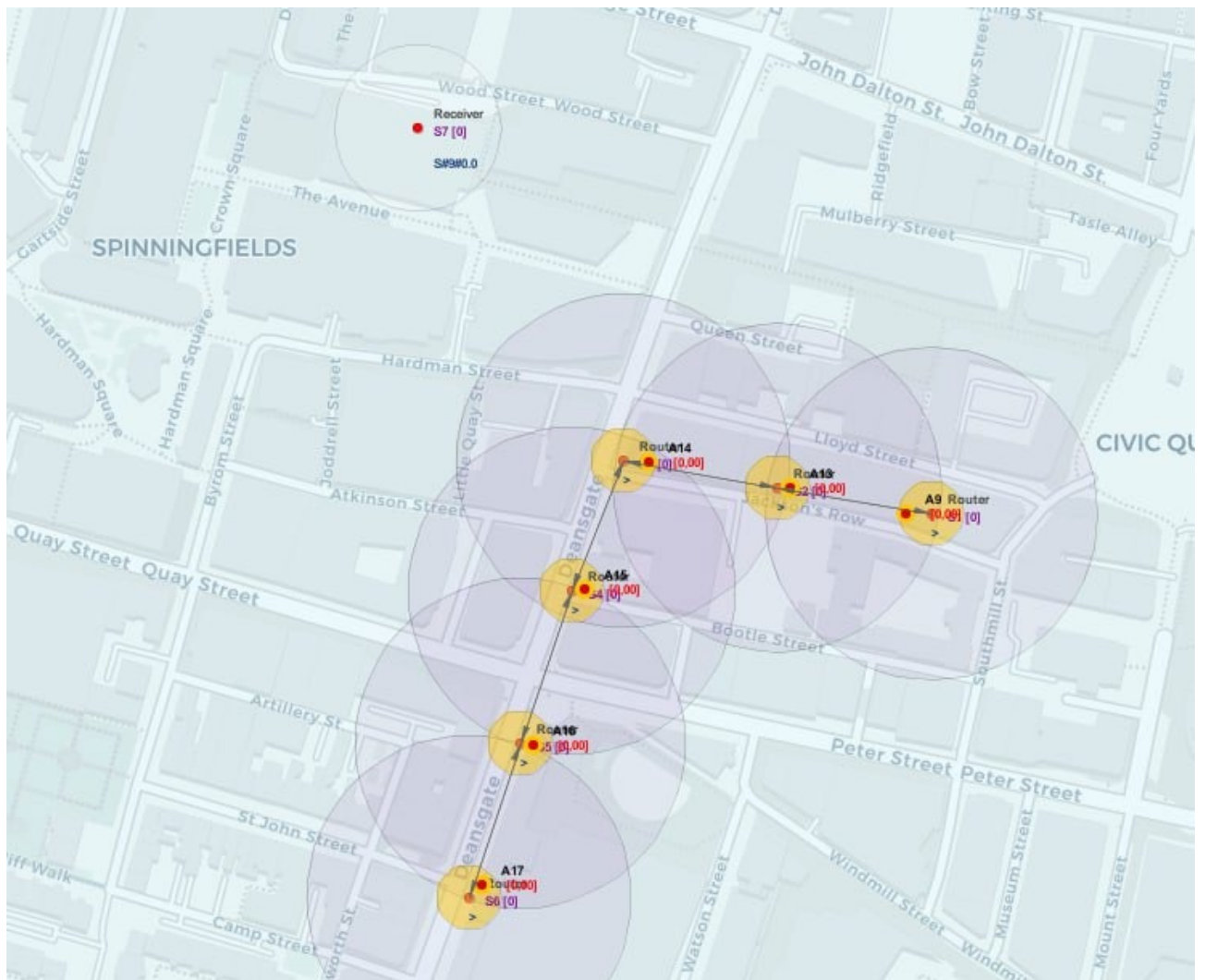


Рис. 2.1. Модель системи в CupCarbon

Роутери в даній моделі обмінюються інформацією протоколом Zigbee, коли вони помічають зміну у інтенсивності світла вони перемикаються на протоколи LoRa та передають інформацію на сервер (рис. 2.2).



Рис. 2.2. Передача даних по протоколу LoRa

2.5 Обґрунтування та організація вхідних та вихідних даних програми

Вхідними даними в розроблюваному програмному забезпеченні, є дані що користувач вносить в текстові поля, при виборі перемикачів, натисканні кнопок інтерфейсу. Також до вхідних даних можна віднести:

- файл івенту, який імітує зміну погоди (*.evt);
- файл скрипта (*.csc).

Вихідними даними є повідомлення, що відображаються у вікнах, повідомлення про результат спрацювання команди, а також лог-файли роботи програми, що пишеться в консоль. Деякі вихідні дані можна зберегти у файли відповідних типів, а саме: лог-файл роботи програми (*.log);

Застосування окремих типів файлів для різних функцій програми полегшує роботу та прискорює роботу з нею. Всі ці файли, при бажанні можна відкрити за допомогою стандартного додатка Windows – «Блокнот».

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

«IoT (Інтернет речей) з'явився з новим поколінням Інтернету, який дозволяє використовувати мобільні телефони та спілкуватися пристрої, які підключаються через Інтернет так само, як комп'ютери. Тепер є можливість віддалено контролювати або контролювати будь-який об'єкт, наприклад робот, глюкометр, будинки, автомобілі, Таким чином, можна створювати необмежену кількість додатків щоб покращити наше повсякденне життя, особливо людей з зниженою рухливістю.

Розгортання кількох вузлів IoT є дуже важким, дорогим і тривалим споживання, і може бути неможливим, коли вузли повинні бути часто перепрограмувати або коли вузли розгорнуті в віддалених місцях, наприклад, у різних містах чи країнах.

Таким чином, використання програмного забезпечення для створення, перенастроювання та моніторингу такої мережі дуже практичні та корисні. Перша платформа. Для цієї мети розроблено CupCarbon-Lab. У цьому ми представимо цю платформу, на основі якої середовище симулятора CupCarbon1i які можуть допомогти проектувати та впроваджувати реальні мережі ефективно з меншими зусиллями, час і гроші. Вузли IoT на основі карт Raspberry Pi, можуть бути фізично встановлені та розгорнуті на їх реальному місці після чого можна налаштувати та запрограмувати кожен вузол з платформи CupCarbon» [10].

«21 століття стикається з енергетичною кризою. Енергоефективна система в промислове, комерційне, житлове та вуличне освітлення є найефективнішим ідходом до вирішення або послаблення енергії криза. У будівництві, управлінні та експлуатації енергетики план економії зменшує 20%-30% споживання енергії без зміна конструкції та апаратури енергетики будівлі система живлення.

Вуличне освітлення є однією з найважливіших характеристик міста, забезпечуючи безпечніші вулиці, парки та центри міста. Вони споживають Величезна частина міського бюджету становить 40% електроенергії рахунки та

значні ресурси для обслуговування та експлуатації. Розумні системи вуличного освітлення закладають основу для становлення розумне місто за рахунок зменшення використання електроенергії, витрат і викидів вуглецю викиди при покращенні середовища. Тому там це величезний потенціал для економії енергії вуличного освітлення ефективного функціонування та управління.

Система керування вуличним освітленням вмикає та вимикає вуличні ліхтарі а контролює стан роботи вуличних ліхтарів з метою забезпечення управління освітленням для автомобілів і пішоходів. Система зв'язку по лінії електропередач є ще одним підходом для керування вуличним освітленням, який представлений у цій статті. Щоб підвищити надійність системи керування мережею, встановлено додатковий кабель зв'язку між лампами та центральним контролером, хоча головним недоліком такого типу керування є дороге обладнання. Датчик світла або заданий час забезпечують сигнал живлення для всіх полюсів лампи в системі самоконтролю. Простота - головна перевага системи самоконтролю. Вразливість датчика світла до погодних умов і відсутність середовища зв'язку між стовпом лампи та центральним контролером є головним недоліком цієї системи.

Бездротова система керування забезпечує гнучку структуру, простий монтаж і двосторонню систему зв'язку. У було представлено інтелектуальний план із врахуванням зменшення споживання енергії, вартості обслуговування, викидів парникових газів і збільшення терміну служби лампи.

Протокол DALI є ще одним підходом до проектування системи керування освітленням Smart Street. Вуличне освітлення можна контролювати автономно за допомогою системи бездротового зв'язку, наприклад GPRS/GSM і бездротової сенсорної мережі (WSN). Метод групового прийняття рішень, натхненний соціальними тваринами та комахами, покращує рівень інтелекту системи керування вуличним освітленням. Комунікації за лініями електропередач (PLC) — це важлива мережева технологія керування енергією для додатків Smart Grid, Smart Cities та Smart Buildings. ПЛК має багато переваг перед іншими методами дротового та бездротового зв'язку. ПЛК привабливий, оскільки немає потреби

прокладати додаткові дроти до пристроїв із живленням. PLC також може працювати там, де радіочастота (RF) не може. Наприклад, інтелектуальні лічильники в підвалі підвалу будівлі навряд чи зможуть використовувати радіочастоту для зв'язку з концентратором даних сусідства. З іншого боку, зв'язок ПЛК може проходити через дроти живлення, щоб дістатися до концентратора даних. З цих причин більшість комунальних підприємств у всьому світі обрали PLC для своїх проектів smart grid, а більшість міст обрали PLC для своїх проектів розумного вуличного освітлення. Комунікаційна технологія лінії електропередач Lonworks для вуличного освітлення була представлена в . Обладнання для вуличного освітлення збирало дані про стан роботи вуличних стовпів на основі i.LON Smart Server у поєднанні з його програмним забезпеченням для керування. Основною метою цієї статті є розробка інтелектуальної системи керування вуличним освітленням на основі технології зв'язку по лінії електропередач LonWorks. Система управління вуличним освітленням може працювати в автоматичному режимі, наприклад вмикати/вимикати та затемнювати вуличні ліхтарі відповідно до запропонованого алгоритму оптимізації енергії та терміну служби лампи.»[11]

Для роботи потрібні:

- персональний комп'ютер чи ноутбук з послідовним портом RS232 чи USB,
- перехідник USB-COM в разі відсутності порту RS232;
- клавіатура;
- миша.

2.6.2. Використані програмні засоби

«У випадку інфрачервоного зв'язку між передавачем і приймачем існує з'єднання в межах прямої видимості, але зв'язок через радіочастоти не має з'єднання в межах прямої видимості. Дальність радіочастотного зв'язку дуже велика порівняно з ПЧ-зв'язком.

Що таке радіочастотний модуль. Як впливає з назви, радіочастотний модуль працює на радіочастоті. Цей діапазон частот коливається від 30 кГц до 300 ГГц. У цій радіочастотній системі цифрові дані представлені як варіації амплітуди несучої хвилі. Цей тип модуляції є амплітудною маніпуляцією (ASK).

У Модуль радіочастотного приймача завжди працюватиме в парі, тобто для надсилання та отримання даних йому потрібні передавач і приймач. Передавач може лише надсилати інформацію, а приймач може лише отримувати її, тому дані можуть надсилатися з одного кінця до іншого, а не навпаки.\

Модуль радіочастотного приймача має чотири контакти, а саме Vcc, Dout, лінійний вихід і землю, як показано вище. Вивід Vcc повинен живитися від регульованого джерела живлення 5 В. Робочий струм цього модуля менше 5,5 мА. Виводи Dout і Linear Out замкнуті разом для отримання сигналу 433 МГц з повітря. Потім цей сигнал демодулюється для отримання даних і надсилається через контакт даних» [12].

У даній кваліфікаційній роботі використовується ресівер, який виступає в ролі сервера він приймає сигнал від роутера (рис. 2.1). Приклад скрипта роутера (див. рис. 2.3).

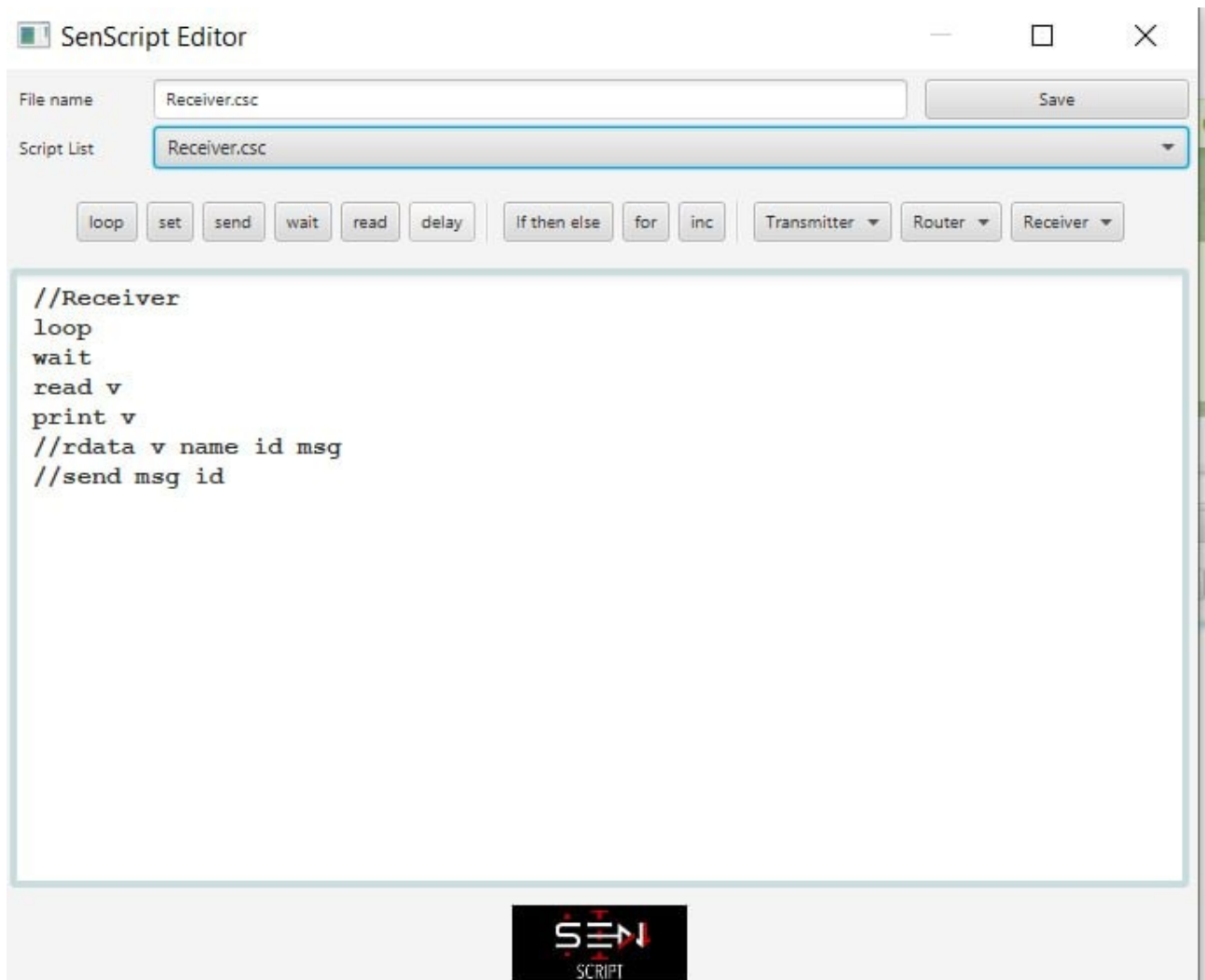


Рис. 2.3. Приклад скрипта ресивера

Які фактори слід враховувати при виборі маршрутизатора IoT. Ключові фактори, які слід враховувати при виборі маршрутизатора IoT, включають швидкість, радіус дії, сумісність із наявними пристроями, функції безпеки, простоту налаштування та використання та додаткові функції, такі як батьківський контроль або QoS.

Чим маршрутизатор IoT відрізняється від звичайного маршрутизатора. Маршрутизатори IoT розроблені для задоволення унікальних вимог пристроїв IoT, таких як низьке енергоспоживання та швидкість передачі даних, а також розширені функції безпеки для захисту пристроїв IoT від кіберзагроз.

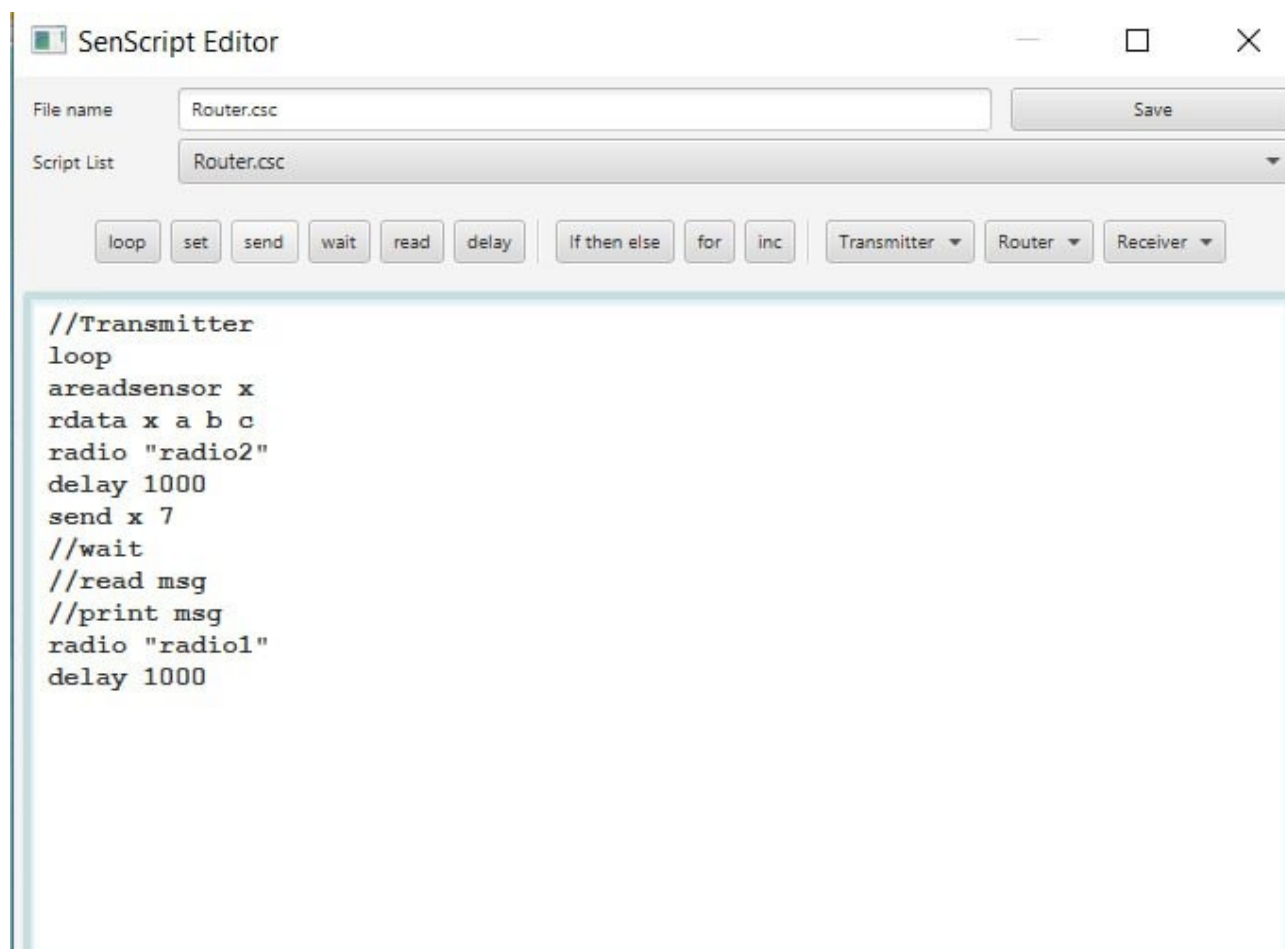
Наскільки важливими є функції безпеки при виборі маршрутизатора IoT. Функції безпеки є дуже важливими при виборі маршрутизатора IoT, оскільки

пристрої IoT можуть бути вразливими до злому та інших загроз безпеці. Шукайте маршрутизатори з такими функціями, як шифрування WPA3, брандмауери та автоматичне оновлення мікропрограми, щоб захистити ваші пристрої та мережу.

У такій мережі перший сенсорний вузол (S1) виконує функцію передавача, другий (S2) – маршрутизатора (ретранслятора), а третій (S3) – приймача. Тобто, вузол маршрутизації інформаційних повідомлень повинен одночасно виконувати функції приймання та відправлення інформаційних повідомлень. Сенсорні вузли приймача та передавача виконують функції аналогічні тим, що були досліджені в практичній» [13].

Роутери зчитують дані з аналогових датчиків, далі роутер перемикається на радіо з технологією LoRa, передає своє ім'я ID і дані на ресивер і повертається на Zigbee, на яких вони спілкуються у стандартному режимі

Приклад роботи наведено на рис 2.4 – 2.8.



```
//Transmitter
loop
areadsensor x
rdata x a b c
radio "radio2"
delay 1000
send x 7
//wait
//read msg
//print msg
radio "radio1"
delay 1000
```

Рис. 2.4. Приклад скрипта роутера



Рис. 2.5. Знаходження роутера у стандартному режимі

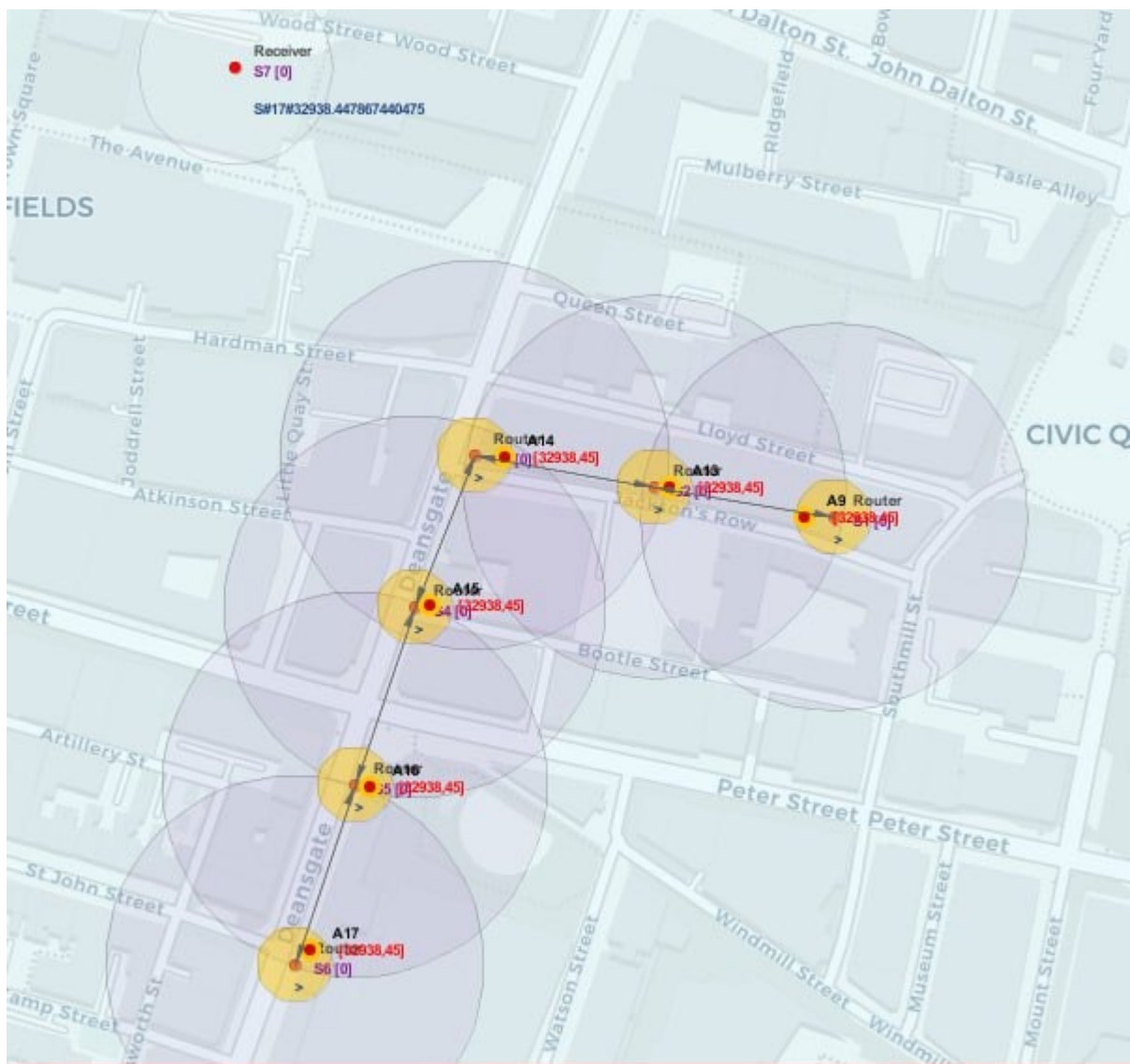


Рис. 2.6. Приклад приймання значення з аналогового датчика



Рис. 2.7. Відсилання повідомлення до приймача



Рис. 2.8. Перехід назад до стандартного режиму

LoRa / ZigBee. «Аналогові датчики перетворюють вхідний сигнал навколишнього середовища у вихідні аналогові сигнали, які є безперервними та змінними. Гарним прикладом аналогових датчиків є термометри, які використовуються в газових водонагрівачах. Індикатор водонагрівача безперервно нагріває термометр. Якщо індикатор гасне, термометр охолоджується, надсилаючи інший аналоговий сигнал, який вказує на те, що подачу газу потрібно припинити». [14]

Аналогові датчики, у данній кваліфікаційній роботі, отримують сигнал навколишнього середовища, який, як відповідає освітленості протягом доби в люксах.

Роль сигнал навколишнього середовища виконує генератор природних подій. Сигнал приймає значення від 111000 до 0,0003 (див. рис. 2.9) [15, 16].

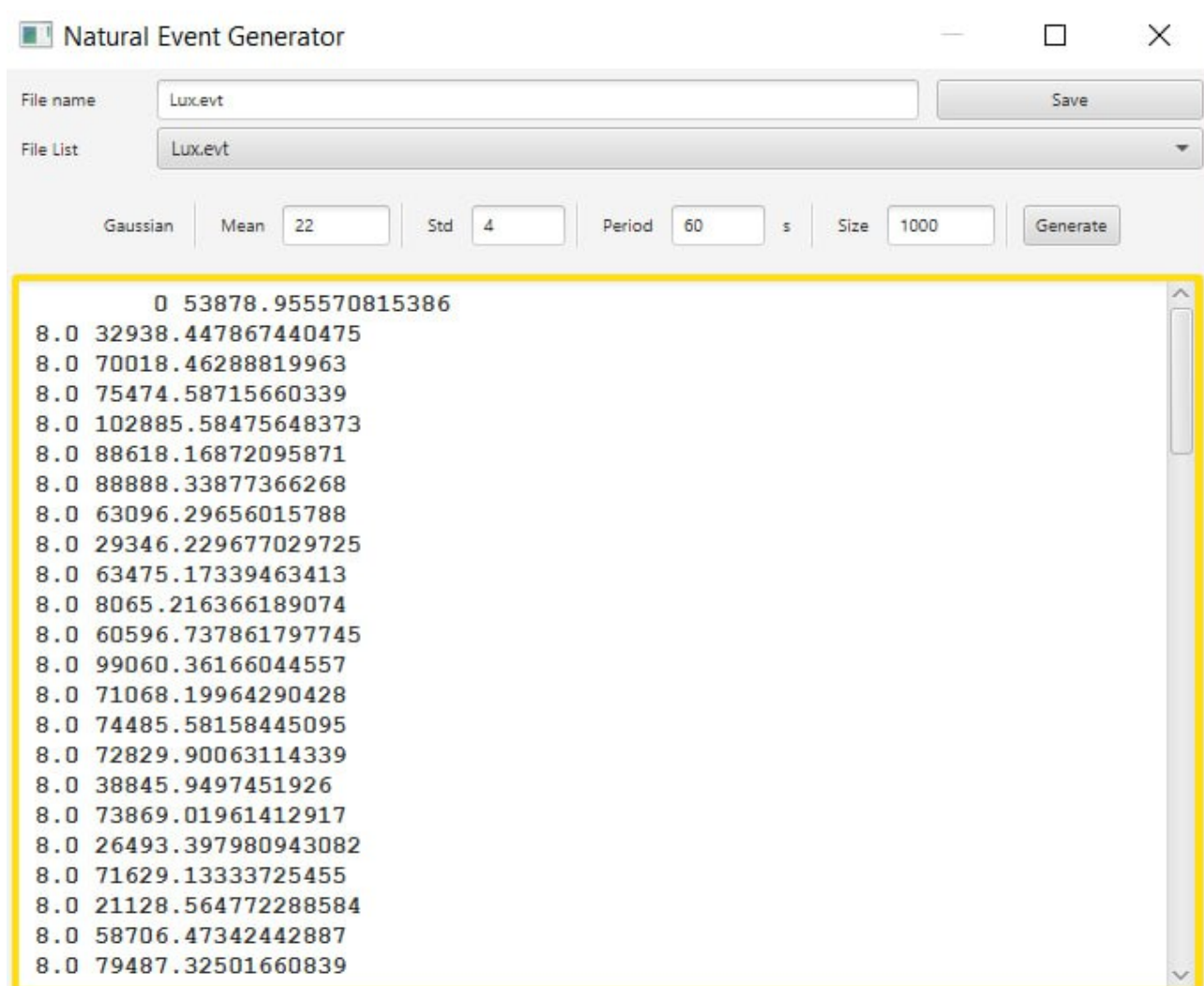


Рис. 2.9. Приклад файлу який генератор природних подій виробляє

У кваліфікаційній роботі були використані протоколи LoRa і Zigbee (рис 2.4 radio “radio 1”, radio “radio2”), але не було пояснення що конкретно вони є і для чого використовуються

У всесвіті Інтернету речей (UIoT) LoRa та його протокол LoRaWAN стали репрезентативною технологією у глобальних мережах з низьким енергоспоживанням (LPWAN). У цій статті обговорюється протокол LoRaWAN, який працює в діапазоні частот нижче гігагерця. Аналіз починається з огляду його діапазону, продуктивності, корисного часу для зарядки, аналізу датчиків,

ефірного часу та порівняння деяких його процесів та продуктивності. З іншого боку, до поняття енергії підходять поверхово, але з великим інтересом для майбутньої роботи, оскільки це одна з найбільш коментованих тем, що викликала особливий інтерес до цього типу технологій» [17].

«LoRa — це техніка бездротової модуляції, яка походить від технології Chirp Spread Spectrum (CSS). Він кодує інформацію на радіохвилях за допомогою імпульсів чирпу - подібно до того, як спілкуються дельфіни та кажани! Модульована передача LoRa стійка до перешкод і може прийматися на великих відстанях.

Модуляцію LoRa та технологію Chirp Spread Spectrum легко зрозуміти на практиці.

LoRa ідеально підходить для додатків, які передають невеликі порції даних з низькою швидкістю передачі даних. Дані можна передавати на більшій відстані порівняно з такими технологіями, як WiFi, Bluetooth або ZigBee. Завдяки цим функціям LoRa добре підходить для датчиків і приводів, які працюють у режимі низької потужності.

LoRa може працювати на безліцензійних субгігагерцевих діапазонах, наприклад, 915 МГц, 868 МГц і 433 МГц. Він також може працювати на частоті 2,4 ГГц, щоб досягти вищої швидкості передачі даних порівняно з субгігагерцевими діапазонами за рахунок діапазону. Ці частоти належать до діапазонів ISM, які зарезервовані на міжнародному рівні для промислових, наукових і медичних цілей.

InflationLoRa — це техніка бездротової модуляції, яка походить від технології Chirp Spread Spectrum (CSS). Він кодує інформацію на радіохвилях за допомогою імпульсів чирпу - подібно до того, як спілкуються дельфіни та кажани! Модульована передача LoRa стійка до перешкод і може прийматися на великих відстанях.

Не лякайтеся складних термінів; Модуляцію LoRa та технологію Chirp Spread Spectrum легко зрозуміти на практиці. Якщо вам цікаво, у цьому відео Річард Веннер пояснює, як працює технологія Chirp Spread Spectrum:

LoRa ідеально підходить для додатків, які передають невеликі порції даних з низькою швидкістю передачі даних. Дані можна передавати на більшій відстані порівняно з такими технологіями, як WiFi, Bluetooth або ZigBee. Завдяки цим функціям LoRa добре підходить для датчиків і приводів, які працюють у режимі низької потужності.

LoRa може працювати на безліцензійних субгігагерцевих діапазонах, наприклад, 915 МГц, 868 МГц і 433 МГц. Він також може працювати на частоті 2,4 ГГц, щоб досягти вищої швидкості передачі даних порівняно з субгігагерцевими діапазонами за рахунок діапазону. Ці частоти належать до діапазонів ISM, які зарезервовані на міжнародному рівні для промислових, наукових і медичних цілей.

LoRaWAN — це протокол рівня керування доступом до медіа (MAC), створений на основі модуляції LoRa. Це програмний рівень, який визначає, як пристрої використовують обладнання LoRa, наприклад, коли вони передають, і формат повідомлень.

LoRaWAN підходить для передачі корисного навантаження невеликого розміру (наприклад, даних датчиків) на великі відстані. Модуляція LoRa забезпечує значно більший діапазон зв'язку з низькою пропускнуою здатністю, ніж інші конкуруючі технології бездротової передачі даних. На наступному малюнку показано деякі технології доступу, які можна використовувати для бездротової передачі даних, і їхні очікувані діапазони передачі в залежності від пропускнуої здатності.

Зазначимо декілька переваг використання протоколу LoRaWAN. Надзвичайно низька потужність — кінцеві пристрої LoRaWAN оптимізовані для роботи в режимі низької потужності та можуть працювати до 10 років від однієї батареї в формі монети.

Велика дальність — шлюзи LoRaWAN можуть передавати та приймати сигнали на відстань понад 10 кілометрів у сільській місцевості та до 3 кілометрів у густонаселених міських районах.

Глибоке проникнення в приміщення - мережі LoRaWAN можуть забезпечити глибоке покриття всередині приміщень і легко покривати багатоповерхові будівлі.

Геолокація – мережа LoRaWAN може визначати місцезнаходження кінцевих пристроїв за допомогою триангуляції без необхідності GPS. Кінцевий пристрій LoRa можна знайти, якщо принаймні три шлюзи приймають його сигнал.

Висока пропускна здатність - мережеві сервери LoRaWAN обробляють мільйони повідомлень від тисяч шлюзів.

Загальнодоступні та приватні розгортання. Легко розгортати публічні та приватні мережі LoRaWAN, використовуючи однакове апаратне забезпечення (шлюзи, кінцеві пристрої, антени) та програмне забезпечення (пересилання пакетів UDP, програмне забезпечення базової станції, стеки LoRaWAN для кінцевих пристроїв).

Наскрізна безпека - LoRaWAN забезпечує безпечний зв'язок між кінцевим пристроєм і сервером додатків за допомогою шифрування AES-128.

Роумінг. Кінцеві пристрої LoRaWAN можуть здійснювати плавне переключення з однієї мережі в іншу.

Аналіз проводиться за допомогою платформи CupCarbon, оскільки вона має дуже зручне графічне середовище, а його кінцевий результат відповідає тому, що потрібний для даного типу технології

Програма сертифікації. Програма сертифікації LoRa Alliance сертифікує кінцеві пристрої та надає кінцевим користувачам впевненість у надійності пристроїв і сумісності зі специфікацією LoRaWAN.

Екосистема – LoRaWAN має дуже велику екосистему виробників пристроїв, виробників шлюзів, виробників антен, постачальників мережевих послуг і розробників програм.

Аналіз проводиться за допомогою платформи CupCarbon, оскільки вона має дуже зручне графічне середовище, а його кінцевий результат відповідає тому, що потрібний для даного типу технології [18].

«ZigBee – це робоча група персональної мережі з низькошвидкісною робочою групою 4. Це технологія домашньої мережі. ZigBee – це технологічний стандарт, створений для контролю та виявлення мережі. Як ми знаємо, ZigBee є персональною мережею групи завдань 4, тому вона заснована на IEEE 802.15.4 та створена Zigbee Alliance.

ZigBee – це відкритий глобальний пакетний протокол, розроблений для забезпечення простої у використанні архітектури безпечних, надійних бездротових мереж із низьким енергоспоживанням. Устаткування управління потоком чи технологічним процесом може бути розміщено будь-де і у своїй підтримувати зв'язок з рештою системи. Його також можна переміщати, оскільки мережі не важливе фізичне розташування датчика, насоса або клапана. Особливості Zigbee:

1. Стохастична адресація: пристрою призначається випадкова адреса та оголошується. Механізм вирішення конфлікту адрес. Батьківському вузлу не потрібно підтримувати призначену таблицю адрес.

2. Керування зв'язками: кожен вузол підтримує якість зв'язків із сусідами. Якість зв'язку використовується як вартість зв'язку під час маршрутизації.

3. Гнучкість частоти: Вузли зазнають перешкод звітують диспетчеру каналів, який потім вибирає інший канал

4. Асиметричне з'єднання: кожен вузол має різну потужність передачі та чутливість. Доріжки можуть бути асиметричними.

5. Керування живленням: маршрутизатори та координатори використовують основне живлення. Кінцеві пристрої використовують батареї.

Переваги Zigbee. Призначений для низького енергоспоживання.

Забезпечує мережеву безпеку та послуги підтримки програм, що працюють на основі IEEE.

Zigbee дозволяє повністю об'єднати в мережу будинки, де всі пристрої можуть спілкуватися та перебувати

Використання в розумному будинку

Легка реалізація

Адекватні функції безпеки.

Низька вартість: мікросхеми та модулі Zigbee відносно недорогі, що робить їх економічно ефективним рішенням для програм IoT.

Сітчаста мережа: Zigbee використовує топологію сітчастої мережі, яка дозволяє пристроям спілкуватися один з одним без необхідності використання центрального концентратора чи маршрутизатора. Це робить його ідеальним для використання в додатках розумного дому, де пристрої повинні спілкуватися один з одним і з центральним центром керування.

Надійність: протокол Zigbee розроблений як високонадійний із надійними механізмами, які забезпечують надійну доставку даних навіть за несприятливих умов.

Недоліки Zigbee. Обмежений радіус дії: Zigbee має відносно короткий діапазон порівняно з іншими протоколами бездротового зв'язку, що може зробити його менш придатним для певних типів додатків або для використання у великих будівлях.

Обмежена швидкість передачі даних: Zigbee розроблено для додатків із низькою швидкістю передачі даних, що може зробити його менш придатним для програм, які потребують високошвидкісної передачі даних.

Сумісність: Zigbee не так широко поширений, як інші протоколи IoT, що може ускладнити пошук пристроїв, сумісних один з одним.

Безпека. Функції безпеки Zigbee не такі надійні, як інші протоколи IoT, що робить його вразливішим до хакерських атак та інших загроз безпеці.

Топології мережі Zigbee:

Зоряна топологія (ZigBee Smart Energy): складається з координатора та кількох кінцевих пристроїв, кінцеві пристрої спілкуються лише з координатором.

Меш-топологія (процес самовідновлення): сітчаста топологія складається з одного координатора, кількох маршрутизаторів і кінцевих пристроїв.

Топологія дерева: у цій топології мережа складається з центрального вузла, який є координатором, кількох маршрутизаторів і кінцевих пристроїв. функція маршрутизатора полягає в розширенні покриття мережі.

Архітектура Zigbee:

Архітектура Zigbee - це комбінація 6 рівнів.

Рівень програми

Рівень інтерфейсу програми

Рівень безпеки

Мережевий рівень

Рівень контролю доступу до середовища

Фізичний рівень

Фізичний рівень: два найнижчі рівні, тобто фізичний і рівень MAC (Medium Access Control), визначаються специфікаціями IEEE 802.15.4. Фізичний рівень є найближчим до апаратного забезпечення та безпосередньо контролює та спілкується з радіо Zigbee. Фізичний рівень перетворює пакети даних у біти по повітрю для передачі та навпаки під час отримання.

Рівень керування доступом до середовища (MAC-рівень): рівень відповідає за інтерфейс між фізичним і мережевим рівнями. Рівень MAC також відповідає за надання ідентифікатора PAN, а також за виявлення мережі через запити маяків.

Мережевий рівень: цей рівень діє як інтерфейс між рівнем MAC і прикладним рівнем. Він відповідає за сітчасту мережу.

Рівень програми: Рівень програми в стеку Zigbee є найвищим рівнем протоколу, і він складається з підрівня підтримки програми та об'єкта пристрою Zigbee. Він містить програми, визначені виробником.

Доступ до каналу:

Метод на основі конфлікту (множинний доступ із визначенням несучої з механізмом уникнення зіткнень)

Безконфліктний метод (координатор виділяє певний часовий інтервал для кожного пристрою (гарантований часовий інтервал (GTS)) [19].

2.6.3. Виклик та завантаження програми

Безпосередньо після завантаження програми cupcarbon та моделі користувач повинен перейти до командного рядка та за допомогою команди cd перейти до теки знаходження програми cupcarbon (cd C:\Users\Achem\Desktop\cc) після треба виконати команду java -jar cupcarbon.jar Після запуску програми, користувач обирає у меню зверху Project->Open Project та вибирає файл у текі проекту з розширенням .cup. Далі щоб запустити проект Simulation->Run Simulation.

Завантажити модель можна з GitHub за посиланням <https://github.com/DeltaTerran/graduate-work> [20]

2.6.4. Опис інтерфейсу користувача

Щоб виконати CupCarbon (файл jar), скористайтеся командним вікном і перейдіть до каталогу, в якому знаходиться файл jar. Потім виконайте таку команду:

java -jar CupCarbon.jar У разі наявності проксі використовуйте таку команду:

java -jar CupCarbon.jar proxy_host_name proxy_number_of_port, як показано на рис. 2.10, графічний інтерфейс користувача (GUI) CupCarbon складається з наступних п'яти основних частин:

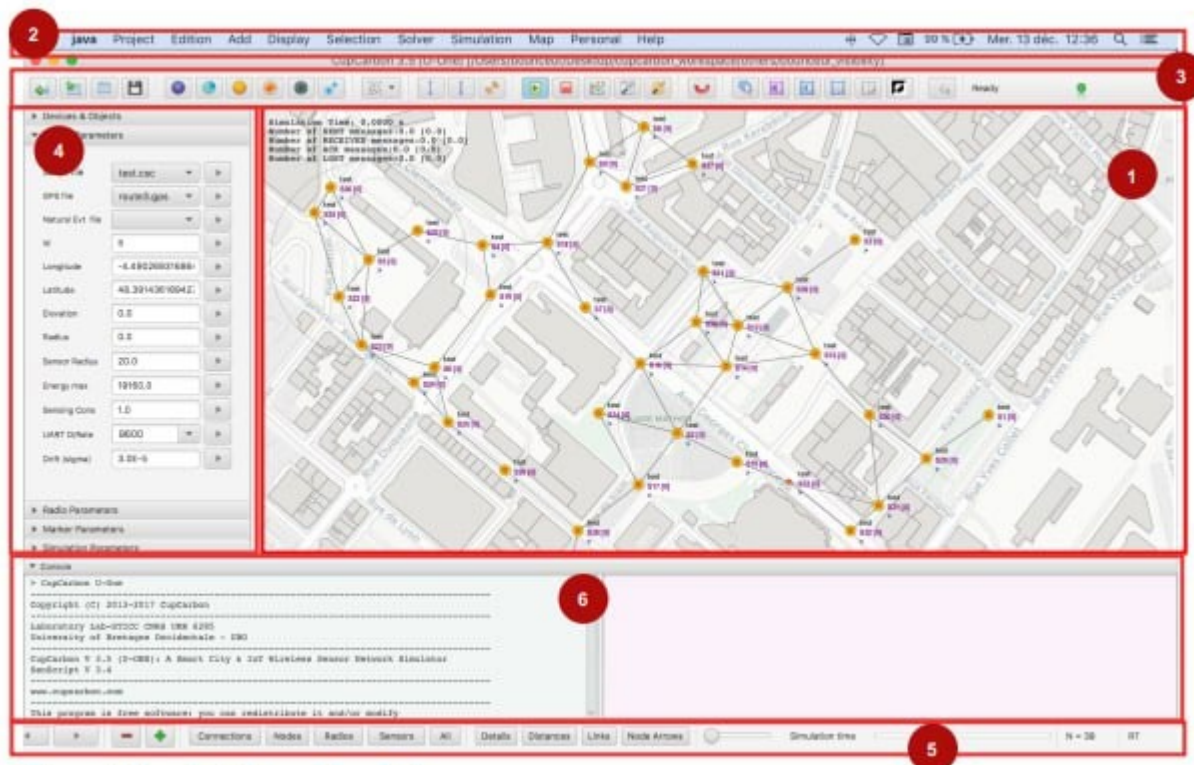


Рис. 2.10. графічний інтерфейс користувача

1. Карта (у центрі)
2. Рядок меню (вгорі)
3. Панель інструментів (під меню)
4. Меню параметрів (ліворуч)
5. Панель стану (внизу)

Меню симуляції. Меню Simulation містить такі підпункти:

1. Запустити симуляцію: щоб почати моделювання
2. Зупинити моделювання: для зупинки моделювання
3. Вікно SenScript: щоб відкрити вікно SenScript
4. Генератор природних подій: для створення значень природних подій.

Цей генератор дозволяє генерувати значення, які можуть бути виявлені сенсорними вузлами, наприклад, температуру, вологість, газ/забруднення тощо. Його також можна використовувати для генерації погодних температур.

РОЗДІЛ 3.

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_\delta, \text{ людино-годин,} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

t_{oml} – витрати праці на налагодження програми на ЕОМ;

t_δ – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C(1 + p), \text{ де} \quad (3.2)$$

q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 200 \cdot 1,3 \cdot (1 + 0,1) = 286.$$

Витрати праці на вивчення опису задачі та визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85)K}, \text{ людино-годин,} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі, $B=1.2 \dots 1.5$;

K – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. до 2 – 1,1.

$$t_u = \frac{286 \cdot 1,45}{85 \cdot 1,1} = 4,43, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t = \frac{Q}{(20 \dots 25)K} \quad (3.4)$$

$$t_a = \frac{286}{20 \cdot 0,8} = 57,75, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25)K} \quad (3.5)$$

$$t_n = \frac{286}{25 \cdot 1,1} = 10,4, \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4...5)K} \quad (3.6)$$

$$t_{\text{отл}} = \frac{286}{5 \cdot 1,1} = 52, \text{ людино-годин,}$$

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,2 \cdot t_{\text{отл}} \quad (3.7)$$

$$t_{\text{отл}}^k = 1,2 \cdot 52 = 62,4, \text{ людино-годин}$$

Витрати праці на підготовку документації:

Витрати праці на підготовку документації:

$$t_d = t_{dp} + t_{do} \quad (3.8)$$

де t_{dp} – трудомісткість підготовки матеріалів і рукопису

$$t_{dp} = \frac{Q}{(15...20)K} \quad (3.9)$$

$$t_{\partial p} = \frac{286}{20 \cdot 1,1} = 13, \text{ людино-годин.}$$

$t_{\partial o}$ – трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0,75 \cdot t_{\partial p} \quad (3.10)$$

$$t_{\partial o} = 0,75 \cdot 13 = 9,75, \text{ людино-годин.}$$

$$t_{\partial} = 13 + 9,75 = 101,06, \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 4,3 + 57,75 + 10,4 + 62,4 + 22,7 = 207,55, \text{ людино-годин.}$$

У результаті ми розрахували, що в загальній складності необхідно 207,55 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ K_{no} включають витрати на заробітну плату виконавця програми Z_{zn} і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{no} = Z_{zn} + Z_{mv}, \text{ грн,} \quad (3.11)$$

де Z_{zn} – заробітна плата виконавців, яка визначається за формулою:

$$Z_{zn} = t \cdot C_{np}, \text{ грн,} \quad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$C_{ПР}$ – середня годинна заробітна плата програміста, грн/година

$$207,55 \cdot 120 = 24906, \text{грн.}$$

$З_{МВ}$ – Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{МВ} = t_{отл} \cdot C_M, \text{ грн,} \quad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$62,4 \cdot 7 = 436,8 \text{ грн.}$$

$$436,8 + 24906 = 25342,8, \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.} \quad (3.14)$$

де B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

$$T = \frac{207,55}{1 \cdot 176} = 1,17 \text{ міс.}$$

Висновки. Час розробки даного програмного забезпечення складає 207,55 людино-годин. Таким чином, очікувана тривалість розробки складе 1,17 місяця

при 40 годинному робочому тижні (місячний фонд робочого часу 176 годин), а витрати на створення програмного забезпечення складатимуть 25342,8 грн.

ВИСНОВОК

В результаті виконання кваліфікаційної роботи та опрацюванні додаткової літератури, отримані наступні висновки:

Вуличні ліхтарі допомагають забезпечити безпеку нічним працівникам, додають краси та яскравості нічному вигляду місцевості та багато інших корисних причин

Вуличне освітлення має важливе значення, починаючи від штучного збільшення тривалості світла і закінчуючи забезпеченням безпеки; мати безпечне водіння вночі.

Правильне освітлення має забезпечувати рівномірний розподіл світлового потоку. Навіть чітко спрямоване світло не повинно мати окремих променів. При попаданні такого освітлення на сітківку ока можна отримати травму, тимчасове погіршення зору, больові відчуття

Використання розробленої моделі дозволить:

- скоротити затрати на реалізацію моделі та подалі випробування;
- за рахунок використання розробленої моделі можуть бути значно зменшені праце-часові витрати під час проєктування інтелектуалізованих систем вуличного освітлення.

Переваги розробленої програми:

- можливість запуску на слабких комп'ютерах;
- легкість освоєння, завдяки доступному інтерфейсу;
- можливість подальшого вдосконалення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Історія інтернету речей / URL: <https://dmkpress.com/files/PDF/978-5-97060-672-8.pdf> (дата звернення 18.02.2023).
2. Кевін Ештон / URL: <https://ru.djvu.online/file/Y3swet5oDKmXS> (дата звернення 25.02.2023).
3. Smart City? – определение / URL: <https://www.twi-global.com/technical-knowledge/faqs/what-is-a-smart-city> (дата звернення 01.07.2023).
4. Як кольорове освітлення (температура світла) впливає на самопочуття / URL: <https://lprof.com/iak-kolorove-osvitlennia-temperatura-svitla-vplyvaie-na-samorochuttia/> (дата звернення 06.02.2023).
5. Що таке IFTTT? Як користуватися послугами If This, Then That / URL: <https://www.computerworld.com/article/3239304/what-is-ifttt-how-to-use-if-this-then-that-services.html> (дата звернення 07.02.2023).
6. Використовуйте розумне освітлення, щоб сповіщати вас про зміни погоди / URL: <https://www.cnet.com/home/smart-home/use-smart-lights-to-notify-you-of-changes-in-weather/> (дата звернення 12.02.2023).
7. Світлодіодна стрічка / URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B2%D1%96%D1%82%D0%BB%D0%BE%D0%B4%D1%96%D0%BE%D0%B4%D0%BD%D0%B0_%D1%81%D1%82%D1%80%D1%96%D1%87%D0%BA%D0%B0 (дата звернення 18.02.2023).
8. Навіщо нам потрібні вуличні ліхтарі / URL: <https://kanglight.com/why-do-we-need-street-lights/> (дата звернення 25.02.2023).
9. Посібник користувача CupCarbon / URL: http://freenwork.com/cupcarbon/cupcarbon_user_guide.pdf (дата звернення 05.03.2023).
10. CupCarbon-Lab: емулятор IoT / URL: <https://hal.science/hal-01704422/document> (дата звернення 15.03.2023).

11. Розумна система керування вуличним освітленням для оптимізації енергоспоживання та терміну служби лампи / URL: https://www.researchgate.net/publication/286900381_A_smart_street_lighting_control_system_for_optimization_of_energy_consumption_and_lamp_life(дата звернення 26.03.2023).
12. Що таке радіочастотний передавач і приймач? / URL: <https://robu.in/what-is-rf-transmitter-and-receiver/>(дата звернення 05.04.2023).
13. Порівняння маршрутизаторів IoT: швидкість, радіус дії та сумісність / URL: <https://firstsourcewireless.com/blogs/blog/a-comparison-of-iot-routers-speed-range-and-compatibility>(дата звернення 12.04.2023).
14. Що таке аналоговий сенсор / URL: <https://www.techtarget.com/whatis/definition/sensor#:~:text=Analog%20sensors%20convert%20the%20environmental,light%20continuously%20heats%20the%20thermo couple>(дата звернення 13.04.2023).
15. Освітлення у люках у денну час / URL: https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%BD%D0%BD%D0%B5_%D1%81%D0%B2%D1%96%D1%82%D0%BB%D0%BE(дата звернення 14.04.2023).
16. Освітленість у люках у нічну час/ URL: <https://uk.wikipedia.org/wiki/%D0%9B%D1%8E%D0%BA%D1%81>(дата звернення 27.04.2023).
17. Аналіз протоколів LoRa та LoRaWAN за допомогою Supercarbon / URL: https://link.springer.com/chapter/10.1007/978-3-030-62554-2_26(дата звернення 27.04.2023).
18. Що таке LoRa і LoRaWAN? / URL: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>(дата звернення 18.06.2023).
19. Представлення ZigBee / URL: <https://www.geeksforgeeks.org/introduction-of-zigbee/>(дата звернення 19.06.2023).

20. Посилання на завантаження програми / URL:
<https://github.com/DeltaTerran/graduate-work/tree/main>(дата звернення
19.06.2023).

ДОДАТОК А

Router.csc

```
//Receiver
```

```
loop
```

```
wait
```

```
read v
```

```
print v
```

```
//rdata v name id msg
```

```
//send msg id
```

Receiver.csc

```
//Transmitter
```

```
loop
```

```
areadsensor x
```

```
rdata x a b c
```

```
radio "radio2"
```

```
delay 1000
```

```
send x 7
```

```
//wait
```

```
//read msg
```

```
//print msg
```

```
radio "radio1"
```

```
delay 1000
```

Lux.evt

```
0 53878.955570815386
```

```
8.0 32938.447867440475
```

```
8.0 70018.46288819963
```

```
8.0 75474.58715660339
```

```
8.0 102885.58475648373
```

```
8.0 88618.16872095871
```

8.0 88888.33877366268
8.0 63096.29656015788
8.0 29346.229677029725
8.0 63475.17339463413
8.0 8065.216366189074
8.0 60596.737861797745
8.0 99060.36166044557
8.0 71068.19964290428
8.0 74485.58158445095
8.0 72829.90063114339
8.0 38845.9497451926
8.0 73869.01961412917
8.0 26493.397980943082
8.0 71629.13333725455
8.0 21128.564772288584
8.0 58706.47342442887
8.0 79487.32501660839
8.0 45573.11535004845
8.0 105652.768009508
8.0 39163.38454015722
8.0 49568.545385495025
8.0 48971.738478918975
8.0 40123.04392543444
8.0 41544.716576775245
8.0 41468.59385013684
8.0 60432.77950753113
8.0 100345.526468731
8.0 91231.97914807138
8.0 39382.5794005863
8.0 87863.99358728771

8.0 111000.41444323056
8.0 79461.86806013978
8.0 37637.92598705884
8.0 54229.751229413836
8.0 16427.591814173713
8.0 88407.01042388692
8.0 81516.42934260512
8.0 46078.58185686625
8.0 60815.17949405499
8.0 97573.31212548405
8.0 81131.24820579337
8.0 82669.39945677148
8.0 64894.5871547222
8.0 73701.9122314586
8.0 56158.26918440213
8.0 34402.69011538723
8.0 78759.90183115857
8.0 100950.19578617167
8.0 76792.56043106058
8.0 68011.79223522975
8.0 46764.12368677651
8.0 66195.08338496386
8.0 39774.35014457494
8.0 63040.55855285146
8.0 91156.09584582917
8.0 60231.80854216747
8.0 111000.53476116288
8.0 2090.471020164623
8.0 107306.20595861603
8.0 60890.04650891117

8.0 92262.70471437108
8.0 29219.15435728737
8.0 21541.608205780234
8.0 28845.252390457015
8.0 22086.906875467175
8.0 69352.16846865448
8.0 83722.37210960728
8.0 12450.859183178269
8.0 87428.98864987359
8.0 60880.75014368082
8.0 80707.80866472133
8.0 44119.54847017623
8.0 44458.244464815434
8.0 38752.460021365594
8.0 47513.60399165415
8.0 120118.88416038002
8.0 79242.57300820726
8.0 74929.09375937331
8.0 42701.30064209824
8.0 64899.116657653816
8.0 34798.536484127544
8.0 23146.6346830243
8.0 101252.62756654777
8.0 122814.75661940279
8.0 0.272824688758
8.0 74905.55676378029
8.0 28219.161970613724
8.0 27438.173049446556
8.0 82616.55484025131
8.0 74488.76643878198

8.0 58123.898284384835
8.0 73553.75127641395
8.0 29009.720326708677
8.0 80616.59621999189
8.0 32938.447867440475
8.0 70018.46288819963
8.0 75474.58715660339
8.0 102885.58475648373
8.0 88618.16872095871
8.0 88888.33877366268
8.0 63096.29656015788
8.0 29346.229677029725
8.0 63475.17339463413
8.0 8065.216366189074
8.0 60596.737861797745
8.0 99060.36166044557
8.0 71068.19964290428
8.0 74485.58158445095
8.0 72829.90063114339
8.0 38845.9497451926
8.0 73869.01961412917
8.0 26493.397980943082
8.0 71629.13333725455
8.0 21128.564772288584
8.0 58706.47342442887
8.0 79487.32501660839
8.0 45573.11535004845
8.0 105652.768009508
8.0 39163.38454015722
8.0 49568.545385495025

8.0 48971.738478918975
8.0 40123.04392543444
8.0 41544.716576775245
8.0 41468.59385013684
8.0 60432.77950753113
8.0 100345.526468731
8.0 91231.97914807138
8.0 39382.5794005863
8.0 87863.99358728771
8.0 111000.41444323056
8.0 79461.86806013978
8.0 37637.92598705884
8.0 54229.751229413836
8.0 16427.591814173713
8.0 88407.01042388692
8.0 81516.42934260512
8.0 46078.58185686625
8.0 60815.17949405499
8.0 97573.31212548405
8.0 81131.24820579337
8.0 82669.39945677148
8.0 64894.5871547222
8.0 73701.9122314586
8.0 56158.26918440213
8.0 34402.69011538723
8.0 78759.90183115857
8.0 100950.19578617167
8.0 76792.56043106058
8.0 68011.79223522975
8.0 46764.12368677651

8.0 66195.08338496386
8.0 39774.35014457494
8.0 63040.55855285146
8.0 91156.09584582917
8.0 60231.80854216747
8.0 111000.53476116288
8.0 2090.471020164623
8.0 107306.20595861603
8.0 60890.04650891117
8.0 92262.70471437108
8.0 29219.15435728737
8.0 21541.608205780234
8.0 28845.252390457015
8.0 22086.906875467175
8.0 69352.16846865448
8.0 83722.37210960728
8.0 12450.859183178269
8.0 87428.98864987359
8.0 60880.75014368082
8.0 80707.80866472133
8.0 44119.54847017623
8.0 44458.244464815434
8.0 38752.460021365594

ДОДАТОК Б
ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ДОДАТОК В
ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Марченко.doc	Пояснювальна записка до дипломного проекту. Документ Word.
Диплом_Марченко.pdf	Пояснювальна записка до дипломного проекту в форматі PDF
Програма	
Дiplom.cup	Файл проекту
Додаткові файли	
Lux.evt	Файл генератора натуральних подій
Презентація	
Презентація Марченко.ppt	Презентація дипломного проекту