

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
УПРАВЛІННЯ ПРИСТРОЯМИ РОЗУМНОГО БУДИНКУ»

Виконав: студент 4 курсу, групи ПД-41
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Гайдайчук А.К.

(прізвище та ініціали)

Керівник _____

Бондарчук А.П.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтролер _____

(прізвище та ініціали)

- 5.4 Структура системи «Розумний будинок»;
- 5.5 Аналіз засобів управління;
- 5.6 Концепція побудови системи управління інтелектуальною будівлею;
- 5.7 Приклад автоматизованих системи управління;
- 5.8 Інформаційні технології систем управління;
- 5.9 Аналіз факторів впливу та алгоритму функціонування систем управління безпекою будинку;
- 5.10 Апаратні засоби та елементи системи;
- 5.11 Зборка елементів системи інтелектуальної будівлі;
- 5.12 Програмування пристроїв;
- 5.13 Висервки.

6. Дата видачі завдання 19.04.2021 р

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1.	Підбір науково-технічної літератури	20.04.2021 р	Виконано
2.	Аналіз сучасного стану систем управління інтелектуальною будівлею	26.04.2021 р	Виконано
3.	Формування вихідних даних та теоретичних основ функціонування ті побудови систем управління розумним будинком	03.05.2021 р	Виконано
4.	Практична реалізація системи управління розумним будинком	16.05.2021 р	Виконано
5.	Розробка висноаків, оформлення роботи	22.05.2021 р	Виконано
6.	Розробка обов`язкових демонстраційних матеріалів	27.05.2021 р	Виконано

Студент _____ Гайдайчук А.К.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Бондарчук А.П.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 66 с., 4 табл., 43 рис., 28 джерел.

ARDUINO, РОЗУМНИЙ БУДИНОК, СИСТЕМА УПРАВЛІННЯ, ДАТЧИК, АВТОМАТИЗАЦІЯ, КОНТРОЛЕР, ІНТЕРФЕЙС, АВАРІЙНІ СИТУАЦІЇ, SCADA.

Об'єкт дослідження – підсистема управління «Розумного будинку»

Предмет дослідження – процес управління безпекою інтелектуального будинку.

Мета роботи – побудова інформаційної системи управління безпеки інтелектуальної будівлі.

Методи дослідження. У бакалаврській дипломній роботі обґрунтовані теоретичними дослідженнями з застосуванням елементів системного аналізу. Використані математичного аналізу, теорії баз даних, аналітико-синтетичної обробки інформації, методи теорії ймовірностей, теорії масового обслуговування.

Використання системи розумного будинку надає можливість управляти всіма датчиками і пристроями в будинку віддалено, з ПК на роботі або зі смартфона в пробці. Веб-додаток дозволяє тримати в курсі, що відбувається вдома в будь-який час в будь-якому місці, надаючи змогу бути людині у більш комфортних умовах.

У роботі сформовано теоретичні основи побудови систем управління розумним будинком яка є основою концепції систем управління безпекою інтелектуальної будівлі. На основі представленої концепції розроблена загальна технологія функціонування систем управління розумним будинком.

. ЗМІСТ

	Стор.
ВСТУП	8
1 АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ	10
1.1 Основні принципи побудови та функціонування системи розумного будинку.....	10
1.2 Структура системи інтелектуальної будівлі.....	12
1.3 Огляд особливостей управління системою розумного будинку.....	13
1.4 Концепція побудови системи управління.....	17
1.5 Приклад автоматизованих системи управління.....	21
2 ТЕОРЕТИЧНІ ОСНОВ ФУНКЦІОНУВАННЯ ТІ ПОБУДОВИ СИСТЕМ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ	26
2.1 Інформаційні технології систем управління розумним будинком.....	26
2.2 Математична модель вибору рішень в типових функціональних підсистемах.....	29
2.3 Аналіз факторів впливу на функціонування системи «Розумний будинок».....	37
2.4 Підсистема збору вхідних даних.....	39
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ «РОЗУМНОГО БУДИНОКУ	48
3.1 Алгоритм функціонування системи управління безпекою будинку....	48
3.2 Вибір апаратних засобів елементів системи.....	49
3.3 Дослідження особливостей зборки елементів системи «Розумний будинок».....	61
3.4 Програмування пристроїв.....	64
ВИСНОВКИ	73
ПЕРЕЛІК ПОСИЛАНЬ	74
ДОДАТКИ	76

ВСТУП

Актуальність теми. Сьогодні складно уявити світ без автоматизації. Житловий будинок не виняток. У звичайному житті в заміському будинку або квартирі виконується величезна кількість дій, які могли б виконуватися без нашої участі. Можна автоматизувати все, від включення світла до управління мікрокліматом в кімнаті. Дана можливість з'явилася завдяки системі «Розумний будинок». Незаперечною перевагою системи розумного будинку є можливість управляти всіма датчиками і пристроями в будинку віддалено, з ПК на роботі або зі смартфона в пробці, веб-додаток дозволяє бути в курсі того, що відбувається вдома в будь-який час в будь-якому місці.

За допомогою таких систем можна здійснювати цілодобовий контроль за будь-яким об'єктом, а саме квартирою, котеджом, офісом або складом. Зараз існує велика кількість таких систем, розроблених як великими компаніями, так і звичайними людьми. Такі системи бувають дуже різноманітними але виконують корисну роль, а саме: захищають від несанкціонованого доступу сторонніх осіб, контроль стану протипожежної безпеки, витік газу та інше.

Об'єкт дослідження – підсистема управління «Розумного будинку»

Предмет дослідження – процес управління безпекою інтелектуального будинку.

Мета роботи – побудова інформаційної системи управління безпеки інтелектуальної будівлі.

Методи дослідження. У бакалаврській дипломній роботі обгрунтовані теоретичними дослідженнями з застосуванням елементів системного аналізу. Використані математичного аналізу, теорії баз даних, аналітико-синтетичної обробки інформації, методи теорії ймовірностей, теорії масового обслуговування.

Для виконання поставленої мети у бакалаврській роботі розроблено та виконано завдання: Аналіз сучасного стану систем управління інтелектуальною будівлею; Формування теоретичних основ побудови систем управління розумним

будинком; Практична реалізація управління систем безпеки інтелектуального будинку.

Джерела дослідження:

<https://smart-home-company.com.ua/gotovye-resheniya/>

<https://www.asutpp.ru/sistema-umnyj-dom.html>

<https://arduinomaster.ru/umnyj-dom/proekty-umnogo-doma/>

Наукова новизна одержаних результатів. Наукова новизна дипломної роботи полягає у розробці інтелектуальної системи управління безпеки будинку на базі мікроконтролеру Arduino.

Практичне значення одержаних результатів. Сформовано теоретичні основи побудови систем управління розумним будинком яка є основою концепції систем управління безпекою інтелектуальної будівлі. На основі представленої концепції розроблена загальна технологія функціонування СУРБ.

1 АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ УПРАВЛІННЯ РОЗУМНИМ БУДИНКОМ

1.1 Основні принципи побудови та функціонування системи розумного будинку

Сучасний «розумний дім» втілює в собі безліч інноваційних розробок, які зробили його унікальним з безпеки і комфортабельності. Наявність всіх цих розробок дозволяє сьогодні втілювати мрії в життя, тепер власнику житла зовсім необов'язково турбуватися про свій будинок, адже він завжди під контролем обладнання, яке не дає збоїв і працює цілодобово весь рік, навіть коли нікого немає в будинку. Принцип роботи системи «Розумний дім» полягає в автоматизації всього, з чого складається житлова споруда: освітлення, кондиціонування, система безпеки, електроенергія, опалення, водопостачання та водовідведення і так далі. До основних підсистем «розумного дому» відносяться: клімат-контроль, освітлення, мультимедіа (аудіо і відео), охоронні системи, зв'язок та інші рис. 1.1. У стандартному проекті «розумного дому» можна виділити три основні підмережі: мережа мультимедійних пристроїв, мережа електроосвітлювального обладнання і сенсорну мережу. В останньому випадку це датчики руху, світла, температури, тиску, вологості, вібрації і т.п. Таким чином, «розумний дім» складається з програмного і апаратного забезпечення, датчиків і проводової/ безпроводової мережі рис. 1.2.

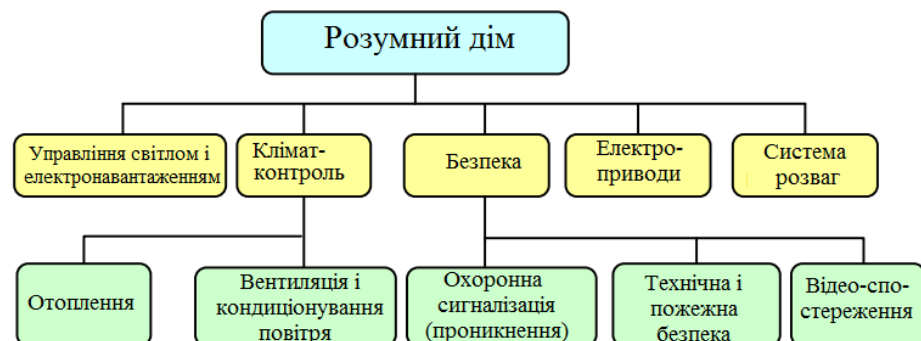


Рисунок 1.1 - Основні підсистеми «розумного дому»

У загальному випадку, «розумний дім» надає його власнику такі переваги:

- 1) зниження споживання ресурсів (газ, вода, електроенергія);
- 2) високий рівень комфорту;
- 3) забезпечення необхідної взаємодії всіх автоматизуються систем об'єкта нерухомості, завдання різних режимів роботи;
- 4) зниження ймовірності виникнення аварійних ситуацій;
- 5) підвищення оперативності, простоти і зручності управління.



Рисунок 1.2 - Основні компоненти розумного будинку

Для автоматизації будинку смарт-вузли можуть бути інтегровані безпосередньо в побутові прилади, наприклад в пилососи, мікрохвильові печі, холодильники і телевізори. Вони можуть взаємодіяти один з одним і з зовнішнім мережею через інтернет. Це дозволить кінцевим користувачам легко управляти пристроями будинку як локально, так і віддалено. Більшість побутових пристроїв з категорії «розумних» речей можна поділити на дві групи за типом використання інтернету. До першої групи належить техніка, яка через WWW оновлює своє програмне забезпечення, отримує нові функції, приймає сигнали від господаря, який знаходиться далеко і, відповідно, відправляє йому інформацію, яка підтверджує виконані дії та свій стан. Цей тип використання інтернету побутової технікою є найбільш розумним і здатний довести потенційному споживачеві свою корисність. До другої групи входить техніка, в якій інтернет є як би стороннім

тілом. Суть рішення в тому, що в абсолютно звичний побутовий прилад, типу мікрохвильовки або холодильника, вбудовується спрощений комп'ютер і дисплей, після чого з їх допомогою можна отримувати мультимедійні розваги там, де їх раніше не було, наприклад, на тій же кухні.

Розумний будинок - це система інтелектуальної автоматики для управління інженерними системами сучасної будівлі.

Будь-якій людині в будинку, в квартирі або в офісі важливо відчувати себе комфортно і в безпеці. Саме ці два завдання плюс естетика зовнішнього вигляду пристроїв - і є основні цільові установки, на які орієнтовані системи «Розумний дім». Інтелектуальна автоматика управляє всіма інженерними системами в будинку, дозволяє людині централізовано встановлювати комфортні для себе - температуру, вологість, освітленість в кімнатах, зонах, і забезпечує безпеку. Система «Розумний дім» включає в себе наступні об'єкти автоматизації: Управління освітленням; Управління електроприводами; Клімат контроль; Управління системою вентиляції;

Централізоване управління системами: домашнього кінотеатру; мультірум; системи відеоспостереження; ОПС (охоронно-пожежна сигналізація); СКД (системи контролю доступу); Контроль навантажень і аварійних станів; Управління інженерним обладнанням з сенсорних панелей; Сервер управління.

Система «Розумний дім» забезпечує механізм централізованого контролю та інтелектуального управління в житлових, офісних або громадських приміщеннях. З інсталяцією подібної системи вдома чи на роботі кожен користувач отримує можливість в рамках загальної середовища проживання задавати параметри власного індивідуального середовища (світло, температура повітря, звук і т.д.), в т.ч. порядок роботи системи.

1.2 Структура системи інтелектуальної будівлі

Структура системи «Розумний будинок» складається з трьох рівнів, а саме нижнього, середнього і верхнього. Нижній рівень, це датчики і виконавчі елементи,

такі як сервоприводи, датчики вологості і температури, ультразвукові сенсори тощо.



Рисунок 1.3 - Структура системи «Розумний будинок»

Середній рівень системи «Розумний будинок» в нашому випадку представлений у вигляді контролера Arduino UNO, до якого приєднуються все датчики і виконавчі елементи. Верхній рівень, це SCADA (в разі промислових систем) або ж Web-додаток (в нашому випадку), через яке здійснюється управління системою «Розумний будинок». На даний момент вже існують SCADA-системи розумного будинку, проте дані системи реалізовані на промислових контролерів, що тягне за собою високу вартість кінцевого продукту. Наша ж система реалізована на базі контролера Arduino UNO, вартість якого доступна для середнього класу споживача.

1.3 Огляд особливостей управління системою розумнобудинку

На даний момент відбувається процес об'єднання мереж, створюється оптимальний інтерфейс між мережею та кінцевим пристроєм і, головне, інтерфейс між людиною і машиною у вигляді графічного призначеного для користувача інтерфейсу, сенсорного екрану або просто ідентифікації голосу або жеста. Основні функції розумного будинку включають в себе управління наступними системами:

1. Інфраструктура життєзабезпечення (управління освітленістю, управління опаленням, управління мікрокліматом);

2. Системи безпеки (датчики руху, сигналізація, блокування вікон і дверей, імітація присутності, оповіщення про аварії в будинку);

3. Битова техніка (управління телевізором, управління холодильником, управління чайником/кавоваркою, підтримка функцій контролю і управління через інтернет);

4. Енергетика (контроль альтернативних джерел енергетики);

5. Обслуговування (автоматичний опитування працездатності автоматизировани приладів, автоматичній опрос працездатності окремих елементів УД, повідомлення про несправності (включаючи відправку повідомлень і листів), моніторинг і управління (включаючи взаємодію через інтернет) .

Види управління: Перший вид управління за профілями мешканців, що мають свої пріоритети або в ручному режимі (за допомогою пульта, сенсорних панелей, смартфона, комп'ютера):

- Управління розумним будинком за допомогою сенсорної панелі. Сенсорная панель управління являє собою один з найпростіших способів управління умним домом. Сенсорна панель представлена на рис. 1.4, забезпечена невеликим рідкокристалічним дисплеєм, використовуючи який можна активувати всі необхідні комунікаційні системи, а також запрограмувати сценарій їх роботи.



Рисунок 1.4 - Сенсорна панель

Для того щоб почати роботу на сенсорній панелі необхідно вибрати віконце з кнопками-елементами управління. Далі здійснюється перемикання між розділами і відбувається настройка інтерфейсу відповідно до ваших смаків і вимог. При цьому можна здійснювати управління, як опціональними групами (наприклад, опалення освітлення), так і всім приміщенням. Можливо налаштувати інтерфейс сенсорної панелі так, щоб на ньому відображалася повна інформація про стан всіх систем, розташувавши її на робочому столі з швидким викликом необхідного вікна. Корисною функцією сенсорної панелі є «блокування від дітей», яка дозволяє запобігти деякі небажані події.

- Управління розумним будинком з дистанційного пульта управління Для того щоб управляти розумним будинком за допомогою дистанційного пульта управління (ДУ), який продемонстровано на рис. 1.5, необхідно підключити всі системи до радіовимикач. Зазвичай пульт дистанційного керування має 16 каналів. При цьому до кожного з них можна підключити одне або кілька пристроїв, за рахунок чого відпадає необхідність використовувати свій пульт для кожного конкретного пристрою окремо. В даному випадку всі системи управляються за допомогою одного єдиного пульта дистанційного керування [3]



Рисунок 1.5 - Пульт дистанційного управління

Пульт дистанційного керування оснащений таймером, який дозволяє запрограмувати роботу пристроїв так, щоб вони включалися і/або вимикалися в

певний час. Однак тут необхідно знати, що роботу пульта дистанційного керування, на відміну від сенсорної панелі, можна налаштувати на кілька днів вперед. Додатково до цього пульт не так наочний, як панель. Управління розумним будинком за допомогою комп'ютера Розумним будинком, можливо управляти за допомогою комп'ютера, ноутбука і навіть кишенькового ПК, який буде виконувати роль сервера. При цьому користувач отримує можливість управляти своїм будинком дистанційно. Для здійснення такого управління комп'ютер необхідно підключити до системи управління з бездротовим або провідним каналам передачі даних. Таким чином, можна виконувати будь-які дії, доступні з сенсорної панелі. Більш того, керувати розумним будинком стає можливим на відстані, використовуючи web-додаток, яке встановлюється на будь-який комп'ютер, що має вихід в інтернет. Управління розумним будинком з мобільного телефону Сучасні мобільні телефони і смартфони представлені на рис.1.6 в більшості випадків підтримують WAP-технології і GPRS-комунікації [11].



Рисунок 1.6 – Смартфон

Управляти розумним будинком можна і за допомогою стаціонарного телефону. Наприклад, використовуючи домофон, можна влаштувати так, що подзвонив у двері людина ні в якому разі не запідозрить, що вас немає вдома. В даному випадку Другий вид управління на базі системи II (на базі промислових контролерів). Як приклад можна навести деякі багатофункціональні контролери, які зображені на рис. 1.7, призначені для управління в системі РБ: а - Ocelot (X10), б - TVLINK 868 GSMAL (Teleco), в - Neroll 8450-50 (Nero). Вони інтегруються в

систему передачі сигналів (зазвичай поєднану з мережевим каналом за технологією powerline) і здійснюють управління-моніторинг активними (пристрої) і пасивними (Датчики) елементами всього РБ.



Рисунок 1.7- Багатофункціональні контролери в корпусному виконанні

Описані пристрої управління пройшли багаторазові впровадження є надійними сертифікованими інструментами реалізації систем УД. Але їх застосування пов'язане з певними обмеженнями, що не дозволяють повноцінно скористатися перевагою концепції УД. Розглянемо ці обмеження: Жорстка стандартизація інтерфейсів пристрою управління; Обмежень число пристроїв, сумісних без доопрацювання з приведеними системами управління; Жорстка стандартизація протоколів обміну даними в пристрої управління; Обмеження, пов'язані з інтегрованими в пристрої управління елементами пам'яті (як ROM, так і RAM); Складність в розгортанні та налаштування, пов'язані із залученням фірм, що спеціалізуються на просуванні подібних систем; Істотне вартість будь-яких компонентів системи (починаючи від контролерів і закінчуючи сполучними кабелями); Примітивні можливості інтелектуального управління.

1.4 Концепція побудови системи управління

Існуючі підходи до побудови СУРБ, незважаючи на піднесення їх як комплексних систем, має ряд існуючих недоліків. По-перше, вони охоплюють лише найбільш важливі сфери діяльності СУРБ; перш за все, сфери життєзабезпечення і безпеки і скоєно не зачіпають сферу взаємовідносин різних

юридичних і фізичних суб'єктів в зоні контролю. По-друге, підсистеми в складі СУРБ розглядаються як практично незалежні, що створюють проблеми при реалізації багатьох заходів, особливо в АС. На кінець, немає окремо виділеної підсистеми або служби, що відповідає за стратегічний розвиток інтелектуального будівлі [12,13]. Таким чином, найбільш серйозним недоліком існуючих концепцій побудови СУРБ є ігнорування суб'єктів і, перш за все, ЛЗ в якості складової частини СУРБ, тобто повноцінне управління інтелектуальною будівлею обов'язково має ґрунтуватися на розгляд цієї будівлі як соціотехнічної системи, гармонійно поєднує в собі будівлю, всі його технічні та інші служби з ЛС.

Під соціотехнічною системою розуміється організована сукупність технічних, програмно-апаратних засобів, а також фізичних осіб, в якій кожен із зазначених компонентів має строго певні рамки і обмеження своєї діяльності.

Виділимо деякі з принципів побудови соціотехнічних систем в рамках СУРБ.

1. Технічні засоби в складі СУРБ не повинні створювати ЛЗ проблем і незручностей для проживання, пересування або перебування в зоні контролю.

2. Технічні засоби не повинні порушувати або створювати проблеми для реалізації конституційних прав ЛЗ, і, перш за все, права на приватне життя та особисту таємницю. Особливо виділимо з цих позицій систему відеоспостереження, яка є джерелом збору різноманітної інформації конфіденційного характеру про всі ЛЗ. Тому, забезпечення дотримання даного принципу в СУРБ є досить складним завданням.

3. Інформація, що стосується приватного життя ЛЗ та інших фізичних і юридичних осіб, повинна оброблятися СУРБ в повній відповідності законодавством.

4. СУРБ не повинна обробляти дані, робити керуючі дії або обмежувати можливості будь-яких осіб, в тому числі і ЛЗ, що знаходяться поза контрольованою зоною.

5. При виникненні конфліктних ситуацій в зоні контролю СУРБ все дані, що стосуються цих конфліктних ситуацій, а також ЛЗ, які брали участь в конфлікті,

повинна оброблятися і надаватися, насамперед, із забезпеченням інтересів ЛЗ, але без порушення законодавства.

6. При виникненні конфліктних ситуацій між ЛЗ в зоні контролю СУРБ, остання повинна максимально ефективно використовувати наявні в СУРБ можливості для максимально швидкого погашення конфлікту з максимальним дотриманням інтересів як ЛЗ, що беруть участь в конфлікті, так і всієї спільноти ЛЗ.

У зв'язку з вищесказаним концепція побудови СУРБ повинні бути змінена з урахуванням вимог щодо формування соціотехнічних систем. Саме, пропонується виконати наступні мінімальні зміни в концепції СУРБ, описаної у введенні рис. 1.8. СУРБ включає сім підсистем [8]:

1. Система життєзабезпечення, що відповідає за безперебійне і ефективне функціонування всіх основних функціональних інфраструктур необхідних для нормальної життєдіяльності ЛЗ; саме, холодного і гарячого водопостачання, каналізації, електро- та газопостачання, опалення, вентиляції та кондиціонування, телефонна мережа, управління мікрокліматом, освітлення, управління ліфтами і ескалаторами, локальна мережа СУРБ, система оперативного радіозв'язку, мультимедіа, що включає інтернет, супутникове, ефірне та кабельне телебачення, радіофікація.

2. Система безпеки, що відповідає за збереження майна в зоні контролю і забезпечення безпеки всіх ЛЗ та інших суб'єктів. Включає наступні підсистеми другого рівня: система контролю конструкції будівлі (тобто збереження всіх будівельних елементів конструкцій і споруд в зоні контролю), пожежної безпеки та пожежогасіння, контролю доступу, охоронної сигналізації, відеоспостереження, блискавкозахисту, цивільної оборони.

3. Система інформаційної підтримки, основним призначенням якої є надання легітимним суб'єктам (в рамках законодавчих, нормативних та інших обмежувальних документів) інформації з усіх потрібних питань, пов'язаних з функціонуванням СУРБ, з будівлею і територією в зоні контролю, а також по ЛЗ і

іншим фізичним і юридичним особам, а також інформування ЛЗ за конкретними справами і питань, а також населення в цілому в зоні контролю СУРБ.

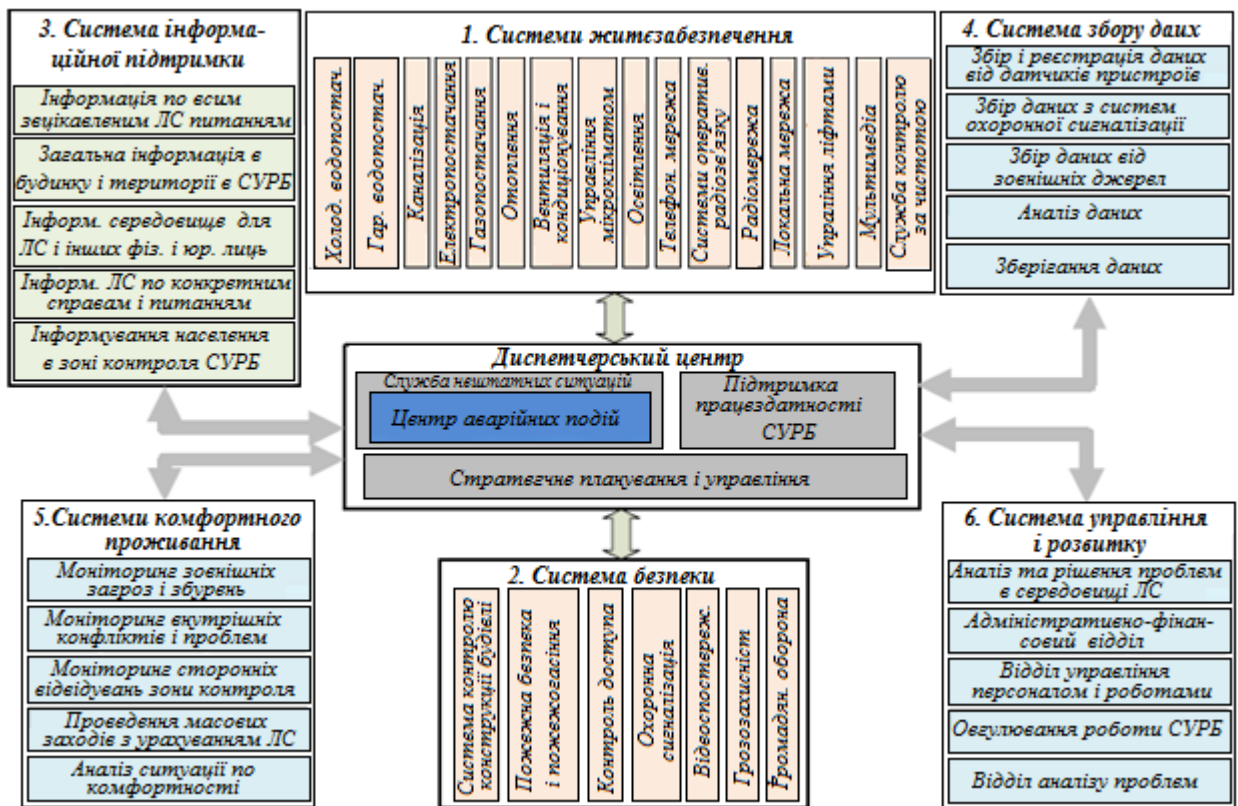


Рисунок 1.8 - Концепція систем управління розумним будинком

4. Система збору та підготовки даних, призначена для забезпечення всіх інших систем СУРБ необхідними за складом і обсягом наборами даних. Включає наступні підсистеми: збору та реєстрації даних від датчиків і датчикових пристроїв, збору даних від систем охоронної сигналізації та відеоспостереження, збір даних із зовнішніх джерел, аналізу даних, зберігання даних.

5. Система забезпечення комфортності, призначена для забезпечення максимальної комфортності перебування всіх ЛЗ в зоні контролю. Включає підсистеми: моніторингу зовнішніх загроз і збурень, моніторингу внутрішніх конфліктів і проблем, моніторингу сторонніх відвідувань зони контролю, проведення масових заходів за участю ЛЗ, аналізу ситуації по комфортності.

6. Система управління та розвитку, призначена для вдосконалення стану всіх споруд в зоні контролю; складається з наступних підсистем: диспетчерський пункт,

комерційної діяльності, аналізу стану споруд і розробка планів по їх реалізації, аналізу та вирішення проблем в середовищі ЛЗ, адміністративно-фінансовий відділ, відділ управління персоналом і роботами, регулювання роботи СУРБ, відділ по АС. Крім перерахованих систем, до складу СУРБ входить безпосередньо керівний центр, до складу якого єдиний диспетчерський центр, який відповідальний за ведення всієї оперативної роботи по підтриманню інтелектуальної будівлі в нормальному стані, коли всі контрольовані параметри всіх підсистем знаходяться в межах встановленого регламенту, а також підсистему стратегічного планування, яка відповідальна за розробку і вирішення стратегічних проблем і перспективних проектів.

1.5 Приклад автоматизованих системи управління

Дослідження програмного рішення моніторингу NetPing. Компанія «Alentis Electronics» є розробником і виробником систем моніторингу навколишнього середовища NetPing. Основна сфера застосування - віддалений контроль і моніторинг пристроїв в будинку і офісі. Задачі, які вирішуються за допомогою пристрою NetPing: Віддалене управління електроживленням; Управління безпекою та відстеження надзвичайних пригод, використовуючи датчики диму, протікання води, витоку газу, антивандальні системи, управління камерами відео спостереження; Управління мікрокліматом за допомогою датчиків температури, вологості і управління кондиціонером через інфрачервоний порт; Управління АТС по порту RS-232; Дистанційне зміна налаштувань в залежності від ситуації; Відправлення повідомлень про неполадки або інші важливі події по засобом SMS, електронна пошта; Доступ до системи в реальному часі через HTTP або SNMP; Управління освітленням та іншими побутовими приладами за розкладом

Пристрої NetPing дозволяють підключити до 16 датчиків на один пристрій. Завдяки вбудованому Web-серверу контроль і управління здійснюється через браузер.

Infrared, Russound, GlobalCache, IRTrans, XBMC, VLC, Samsung SmartTV, panStamps, Denon AVR, Marantz AVR, FreeBox, MythTV, RaZBerry і ін.



Рисунок 1.11 - Інтерфейс програмного забезпечення OpenRemote

Аналіз рішення Home Sapiens. Інтелектуальна система з голосовим управлінням, представляє собою програмне забезпечення. У комплект не входить обладнання, але при цьому забезпечена максимальна сумісність з комп'ютерним «залізом».

Забезпечено інтеграція з системами Z-wave, Gira, ZigBee, x10, C-bus, що дозволить управляти освітлення, побутовою електронікою, системою опалення та ін. Основний упор йде на голосове управління і зручний інтерфейс.

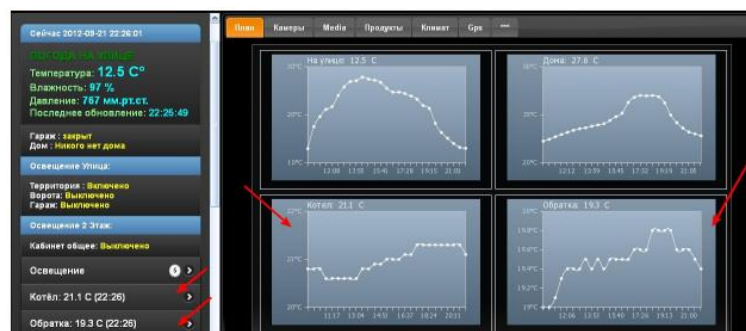


Рисунок 1.12 - Інтерфейс програмного забезпечення Home Sapiens

Особливості програмного рішення MajorDoMo. MajorDoMo - це відкрита програмна платформа, для автоматизації домашніх процесів. Дана система кроссплатформенная і не вимоглива до ресурсів комп'ютера. Може бути

використана, без модулів (датчиків) в якості персонального органайзера. Адачі, які вирішуються за допомогою MajorDoMo: Система безпеки Система мікроклімату; Медіа система; Організатор.

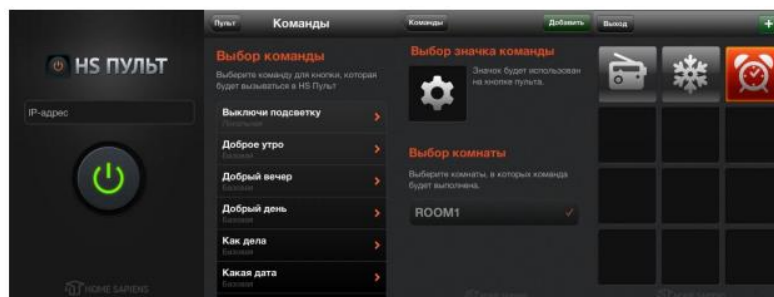


Рисунок 1.13 - Інтерфейс програмного забезпечення MajorDoMo

Програмне рішення Fibaro. Fibaro, система автоматизації будівель заснована на бездротовій технології передачі даних Z-wave. Простий метод монтажу, так як не треба протягувати метри кабелю. Мініатюрні модулі можуть бути встановлені за будь-яким вимикачем світла або в побутовому приладі.

Завдяки бездротовій технології передачі даних пристрою Fibaro можна демонтувати і переносити на нове місце. Система Fibaro постійно сканує систему і при необхідності інформує Вас про подію. Висока інтеграція з іншими системами. МОЗКОМ СИСТЕМИ

Fibaro є Home Center 2. Інтерфейс надає простий контроль над групами пристроїв відповідають за функції - опалення, кондиціонування, освітлення і т.д.



Рисунок 1.14 - Інтерфейс програмного забезпечення Fibaro

Таблиця 1.1- Порівняльні характеристики програмно-апаратних рішень

	NetPing	Open Remote	Home Sapiens	Major DoMo	Fibarо	Arduino
Простота налаштування	+	-	+	+	+	+
Відкритість системи	-	+	-	+	-	+
Мобільні додатки	+	+	+	+	+	+
WEB інтерфейс	+	+	+	+	+	+

З розглянутих готових програмно-апаратних рішень функціонально підходить система Fibarо, так як вона елементарна в налаштування та встановлення додаткового обладнання. Але через високу ціну дана система не підходить.

2 ТЕОРЕТИЧНІ ОСНОВ ФУНКЦІОНУВАННЯ ТА ПОБУДОВИ СИСТЕМ УПРАВЛІННЯ «РОЗУМНИМ БУДИНКОМ»

2.1 Інформаційні технології систем управління розумним будинком

Як впливає з алгоритмів функціонування СУРБ і її деяких компонентів, процеси, пов'язані з управлінням, мають найрізноманітнішу природу, і тому інформаційні технології (ІТ), які покликані підтримувати ці процеси, охоплюють широкий спектр різних програмно-інформаційних засобів [19]. Розглянемо більш детально, які інформаційні технології затребувані в СУРБ.

Початковий етап технології пов'язаний зі збором і реєстрацією даних, що надходять з різних зовнішніх і внутрішніх (бази даних) джерел: від датчиків, систем відеоспостереження, інтернет джерел, від ЛЗ і персоналу, історія перегляду, а також з різних сховищ документів і баз даних (по поточним ситуацій, нормативним і законодавчим документам, по минулим ситуацій і подій, по будівлі і ін.) [20]. Крім того, в процесі обробки дан-них від датчиків можливе використання мікроконтролерів. Нарешті, необхідні засоби контролю коректності, повноти і адекватності.

Таким чином, на етапі збору та підготовки вихідних даних необхідні наступні інформаційні технології:

1. ІТ по обробці та накопичення даних від первинних датчиків в безперервному режимі, що вимірюють різні фізичні параметри, здатні взаємодіяти з мікроконтролерами. Сукупність систем, що забезпечують отримання даних від датчиків і видачу команд на виконавчі пристрої, може ґрунтуватися як на провідних, так і бездротових технологіях. Застосовуються ІТ на основі вбудованих плат збору даних зі стандартним системним інтерфейсом (найбільш поширений інтерфейс PCI), такі CISC, Cypress CY3654. ІТ на основі модулів збору даних із зовнішнім інтерфейсом (RS-232, RS-485, USB), такі як COM Port Toolkit, Cypress CY3654. ІТ на основі цифрових вимірювальних приладів (ЦП) або інтелектуальних датчиків. Для їх організації застосовуються програми з

інтерфейсами: GPIB (IEEE-488), 1-wire, CAN, HART, такі як Hewlett-Packard, VEE 3.0, IAR EWB [25].

2. IT з перманентного, тобто систематичного відповідно до заданого регламенту, пошуку даних в глобальній мережі (в інтернеті), наприклад, програма СайтСпутнік, Resuva.

3. IT з підтримки систем відеоспостереження. Для моніторингу та управління використовується програмне забезпечення для управління системою відеоспостереження і організації автоматизованого робочого місця оператора для моніторингу в реальному часі і роботи з архівом; наприклад, програма WV-ASM200, Secur OS. Для запису і перегляду даних у вигляді зображень від мережевих камер використовується програма ВВ-ННР17 компанії Parasonic, IP Camera Viewer. Для аналітики даних використовується програма WJ-SRV920RU, WV-ASFЕ904 [22].

4. Системи обробки і зберігання документів, наприклад, Zoho Docs, XaitPorter, Process Street.

5. СУБД для зберігання даних різного типу, включаючи схеми, наприклад, програма Profive®.

6. Інформаційні системи контролю якісних характеристик даних різного типу, наприклад, програма master's degree.

Одне з необхідних умов використання різних IT - їх сумісність, наявність можливості їх ефективного використання в рамках єдиної системи.

Другий етап технології пов'язаний з функціонуванням ДЦ, а також підсистеми стратегічного управління. Є серія IT, призначених для управління бізнесом, які можуть бути використані в якості інформаційних технологій функціонування ДЦ і підсистеми стратегічного управління - це ЕАМ-системи. Таким чином, пропонується використовувати:

7. ЕАМ-системи в якості IT підтримки функціонування ДЦ і підсистем стратегічного управління.

Однак, в СУРБ для підтримки функціонування ДЦ доцільніше використовувати SCADA системи, призначені для підтримки функціонування

систем диспетчерського управління та збору даних. Тим самим SCADA системи дозволяють також вирішити проблеми використання ІТ в процесі збору та реєстрації даних, що надходять від датчиків різних типів. Ринок SCADA є одним з найбільш швидко зростаючих ринків систем контролю в світі. Згідно з результатами дослідження глобальної консалтингової компанії Frost & Sullivan «аналіз світового ринку систем SCADA», в 2009 році виручка цього ринку склала 4623,1 млн дол., а до 2016 року, за прогнозами, вона досягне 7074,1 млн дол. [11].

Наступні етапи технології функціонування можуть бути реалізовані в рамках ЕАМ- або SCADA-систем. Доцільно використовувати одночасно обидва типи ІТ: SCADA системи для ДЦ і збору та підготовки даних від датчиків, а ЕАМ-системи перш за все для систем стратегічного управління, а також у складі ДЦ для процесів, пов'язаних з менеджментом.

На етапі вісім виникає необхідність використання ІТ в умовах АС. АС має ряд специфічних особливостей, найбільш важливим з яких є фактор часу, точніше дефіциту часу. Тому використовувані в системах по АС ІТ повинні максимально швидко вирішувати поставлені завдання - навіть, мож-ли, на шкоду якості рішення [2]. Зазначена вимога накладає особливий відбиток на ІТ в системах по АС. В ІТ дозволяють в умовах дефіциту часу отримувати варіанти рішень, які можуть сприяти швидкому вирішенню проблем і які в умовах дефіциту часу вкрай складно отримати іншими методами. Таким чином, приходимо до висновку: використання ІТ в АС вкрай необхідно, але ці технології, в силу своєї специфіки, є оригінали ним типом ІТ. Ключовим етапом в процесі реалізації технології функціонування СУРБ і процедури прийняття рішень є безпосередньо процедура прийняття рішень, що передбачає попередню підготовку варіантів рішень. ІТ, призначені для використання в процесах прийняття рішень, називаються системами підтримки прийняття рішень (СППР).

Ключовим етапом СППР є процедура формування варіантів рішень. Як видно з технології прийняття рішень рис. 2.1

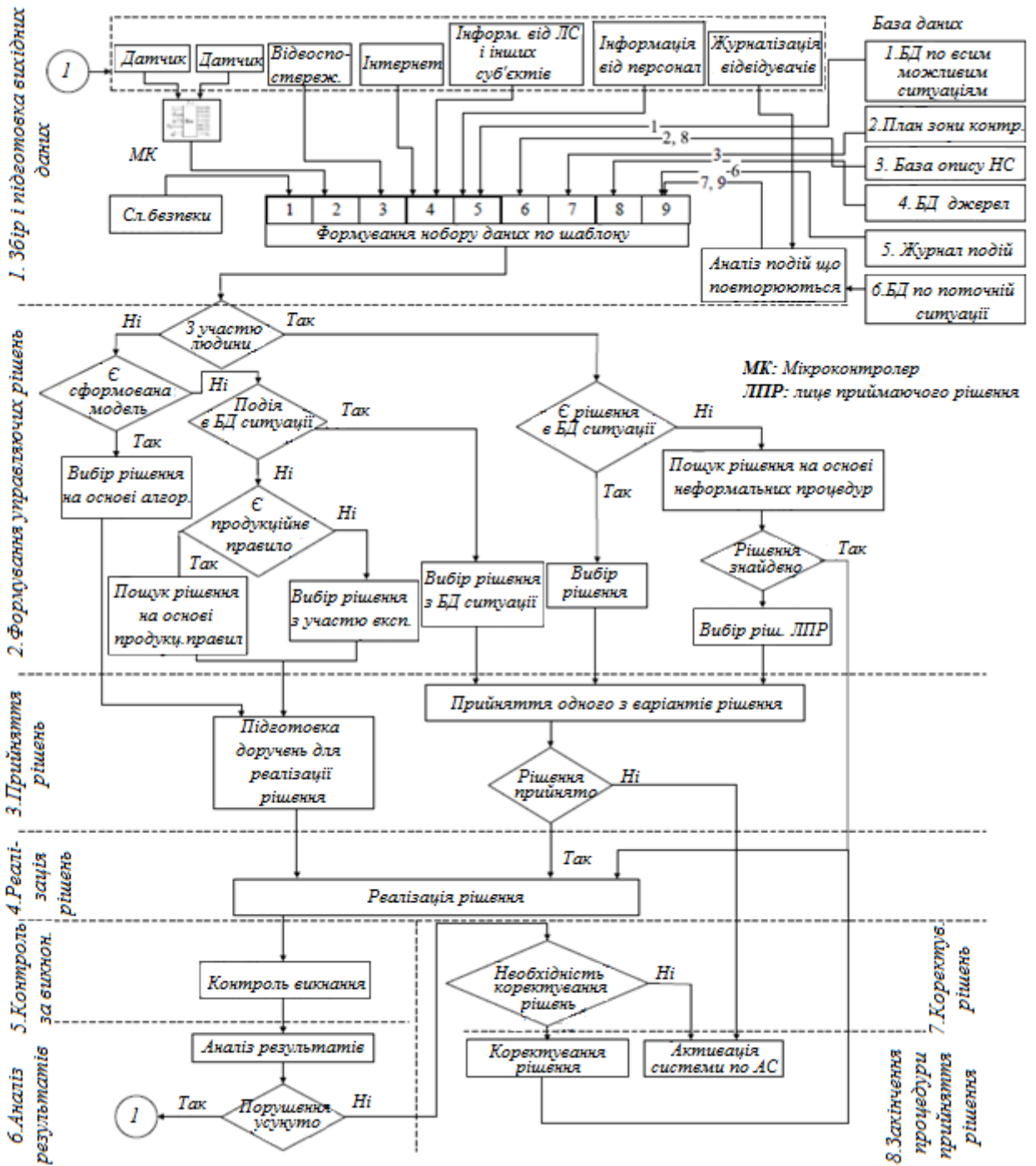


Рисунок 2.1 - Загальний алгоритм процесу прийняття рішень в СУРБ

2.2 Математична модель вибору рішень в типових функціональних підсистемах

Вище була сформована загальна технологія прийняття рішень в СУРБ, де були виділені п'ять типів технологій, які можуть бути затребувані в процесі вибору

варіантів рішень. Оптимальне рішення, однак, може бути отримано, перш за все, на основі першої з цих типів технологій, що спирається на формалізовані моделі. Тому становить інтерес аналіз можливостей використання формалізованих методів стосовно до різних компонентів СУРБ. Нижче ми обмежимося тільки типовими функціональними службами, які характеризуються стандартною схемою прийняття рішень в процесі функціонування; саме, надходження поточні дані від первинних джерел, в переважній більшості, від датчиків різних типів; оцінка відхилення від оптимального або регламентного (тобто встановленого нормативними документами) або переважного для користувача режиму функціонування контрольованого елемента назвемо такий режим регламентним; знаходження за допомогою формалізованих методів оптимальних варіантів (або оптимального варіанту) рішень. Таким чином, типовими компонентами подібних систем є: вихідні дані від первинних джерел, регламентне стан (або набір регламентних станів) системи, набори можливих керуючих впливів. До подібних типовим службам можна віднести майже всі функціональні служби (водо-, газо-, тепло-, електропостачання, каналізації, зв'язку, кондиціонування та ін.).

Для формалізації завдання управління в зазначених системах введемо такі позначення.

Нехай безліч можливих керуючих впливів (рішень системи) $\in D = \{d_1, \dots, d_2\}$, де рішення, N - кількість усіх можливих рішень. Датчикова система характеризується безліччю параметрів $S = \{s_i^j, i = \overline{1, M}; j = \overline{1, K}\}$ (зокрема, тип датчика, значення, його місце розташування, тощо), де K_i - кількість вимірюваних параметрів i -го датчика, j - номер вимірюваного параметра i -го датчика, M - кількість датчиків, доступних службі. Тоді обирається рішення $d_n \in$ функцією всіх параметрів:

$$d_n = f(s_i^j, i = \overline{1, M}, j = \overline{1, K}), \quad (2.1)$$

де безліччю значень функції $f \in D$. Функція $f(.)$ визначається алгоритмом пошуку оптимальних рішень, і в цій якості вона визначає ефективність пошуку оптимальних рішень. Вибір функції $f()$ визначається складом і змістом безлічі

керуючих рішень, особливостями обмежень завдання, особливостями вихідних даних, зокрема, характером їх залежності.

Безпосередньо процес пошуку істотно залежить також від критерію, на основі якого оцінюється близькість поточного варіанту рішення до регламентного станом. У формалізованих моделях критерій пошуку зазвичай задається цільовою функцією. В якості цільових функцій можуть бути використані наступні:

- 1) максимальна близькість параметрів середовища в приміщенні до їх ідеальним (Регламентним) значенням через заданий регламентний проміжок часу;
- 2) мінімізація витрат ресурсів (електроенергії, витрати води, тепла та ін.) В процесі експлуатації будинку. Формалізуємо зазначені цільові функції.

У першому випадку в якості цільової пропонується наступна функція

$$\frac{1}{T} \sum_{t=1}^T \sum_{i=1}^M \sum_{j=1}^{K_i} \omega_{ij} (s_{ionm}^j - s_i^j(t))^2 \rightarrow \min, \quad (2.2)$$

$S_{min} \leq S_i^j(t) \leq S_{max}$ для всіх керованих параметрів. В (1) всі показники S_i^j прив'язані до відповідних моментів часу t , їх зміни задаються співвідношенням

$$s_i^j(t+1) = g_i^j \{s_i^j(t), d_i(t)\} \text{ по всім } i, j, \quad (2.3)$$

де $g_i^j(\cdot)$ - функція, що характеризує засоби підсистем або служб. Мінімізація проводиться за сукупністю всіх рішень $\{d_i(t), t = \overline{1, T}\}$. Можуть бути накладені додаткові обмеження; наприклад, за сумарною витратою енергоресурсів для розглянутих підсистем або служб. Коефіцієнти ω_{ji} задають пріоритетності всіх датчиків, пов'язаних з підсистемами або службами: чим більше ω_{ji} , тим цей датчик важливіше і в великій мірі враховується при оптимізації в (2.1).

Відзначимо ще деякі особливості зазначених функціональних служб. Кожна служба відповідальна зазвичай за один вид обслуговується ресурсу – назовемо його основним ресурсом (гарячу або холодну воду, електроенергію і т.п. Це дозволяє встановити певну (взагалі кажучи, часткову) впорядкованість між можливими варіантами рішень, оскільки переважна кількість можливих варіантів рішень d_i стосуються витрачання основного ресурсу. Саме, рішення d_i «менше» рішення d_j

(записується « $d_i < d_j$ »), якщо можливий обсяг основного ресурсу при вирішенні d_i менше можливого обсягу основного ресурсу при рішенні d_j . Зазначене порівняння стосується також і випадки, коли в рішенні задаються межі зміни основного ресурсу: рішення $d_i < d_j$, якщо інтервали $[a_i; b_i]$ і $[a_j; b_j]$ можливих значень основного ресурсу при рішеннях d_i і d_j задовольняють одну з таких умов: 1) $a_i \leq a_j$ і $b_i \leq b_j$; 2) $a_i \leq a_j$, але $b_i \geq b_j$, так що можливий інтервал змін основного ресурсу в рішенні d_i більше ніж в рішенні d_j . Якщо ж обсяги основних ресурсів в рішеннях d_i і d_j рівні або непорівнянні (наприклад, обмеження на можливі значення основного ресурсу або відсутні, або задані більш складним чином), порівняння йде за іншими параметрами основного ресурсу; наприклад, для гарячої води - по її температурі, напору, кількості домішок; для електроенергії (основний ресурс характеризується перш за все потужністю) - за величиною напруги, стабільності подачі електроенергії, і т.д. Таким чином, на безлічі рішень вводиться модифікація відомого способу упорядкування векторних наборів - лексикографічного впорядкування. При цьому частина рішень (зазвичай, незначна) випадає з введеної системи упорядкування - назовемо ці рішення сингулярними, а їх безліч позначимо через D_s .

Ще однією особливістю даної моделі є наявність часткової впорядкованості на безлічі можливих значень датчиків системи $S = \{s_i^j, i = \overline{1, M}; j = \overline{1, K}\}$. Це обумовлено тим, що всі види ресурсів доставляються до споживачів по гілках мережевої інфраструктури, яка описується впорядкованим деревом (оргдеревом), де напрямок упорядкування відповідає напрямку доставки ресурсу споживачеві або відведення від нього цього ресурсу (наприклад, для систем каналізації, прибирання сміття). Позначимо цю впорядкованість символом « \ll ». Також припускаємо, що нумерація по j показників s_i^j проведена таким чином, що $j = 1$ відповідає обсягу основного ресурсу; $j = 2$ відповідно другого за важливістю показником основного ресурсу; і т.д. Відповідно до впорядкованості оргграфа системи доставки ресурсу упорядковуються і всі датчики, які зазвичай розміщуються в вузлах цього графа або можуть бути віднесені до таких сайтів. Дана впорядкованість накладає певні

обмеження на можливі значення коефіцієнтів ω_{ji} : якщо датчик $s_i < s_k$, то $\omega_{ji} \leq \omega_{kj}$ для всіх j .

Наведемо тепер ще один компонент моделі - можливі обмеження на параметри і змінні моделі. Найбільш важливим системним обмеженням є обмеженість поставляється основного ресурсу за обсягом, що може бути раписано у вигляді наступного нерівності:

$$\sum_{i \in D_v} s_i^1 \leq R_v \quad (2.4)$$

де D_v - сукупність всіх датчиків на кінцевих вузлах орграфа, що описує процес переміщення v -го ресурсу, де безпосередньо і відбувається споживання цього ресурсу (нагадаємо, S_i^1 вказує на обсяг споживання v -го ресурсу в відповідному пункті (вузлі) споживання цього ресурсу), R_v - максимально допустимий обсяг споживання v -го ресурсу в будівлі. Додатково можуть бути встановлені обмеження на обсяг допустимий обсяг споживання кожного з ресурсів в кожному з кінцевих пунктах ((кінцевих вузлах відповідного орграфа v -го ресурсу): $S_i^1 \leq S_{i,max}^1$ для усіх $i \in D_v$. Стосовно конкретних службам можуть обмеження на обсяг споживання ресурсу по окремим галузям графа; наприклад, електроенергії - за окремими гілкам електромережі або на окремі сектори будівлі. Також можуть бути додаткові обмеження за іншими показниками основного ресурсу - не тільки по об'ємним величинам; наприклад, по допустимому діапазону зміни значення гарячої води. Таким чином, сукупність обмежень вимагає уточнення стосовно кожного конкретного споживчого ресурсу в конкретному будинку.

Виберемо тепер можливий метод вирішення отриманої завдання нелінійного програмування. Вибір методу рішення в істотному ступені залежить від аналітичних властивостей функцій $f()$ і $g()$: чим більше гладкими є функції, тим ефективніші можливі методи розв'язання задачі (2.1), (2.2), (2.3); наприклад, в разі диференційованої функції $f()$ і $g()$ можна використовувати методи градієнтного типу, які є ефективними при вирішенні екстремальних задач різного типу. Однак, в загальному випадку функція $f()$ вибору рішення може мати абсолютно різну

природу: задаватися не тільки аналітичним виразами, а й таблицями, продукційними правилами (наприклад, в розглянутій нижче підсистемі ОВК), що ніяк не тягне наявність гладких властивостей у функції $f(\cdot)$. Функція $g(\cdot)$ визначається експлуатаційними характеристиками технічних засобів, які обслуговують конкретних основний ресурс. Наприклад, в підсистемі ОВК, якщо перший показник основного ресурсу описує температуру в приміщенні, другий - швидкість подачі повітря, то функція $g(\cdot)$ в разі прийняття рішення про зміну температури в приміщенні може бути в самому простому випадку описана співвідношеннями:

$$\begin{aligned}
 g_i^1(\cdot) &= s_i^1(t+1) = T_{\text{ном}}(t+1) = \\
 &= \frac{(V_{\text{ном}} - v_n(t) \cdot \sigma_{\text{конд}}) \cdot T_{\text{ном}}(t) + v_n(t) \cdot \sigma_{\text{конд}} \cdot T_n(t)}{V_{\text{ном}}} \quad (2.5) \\
 g_i^2(\cdot) &= s_i^2(t+1) = v_n(t+1) = \varphi_1(d_i(t), P_{\text{дв}}) \\
 g_i^3(\cdot) &= s_i^3(t+1) = T_n(t) = \varphi_2(d_i(t), P_{\text{дв}})
 \end{aligned}$$

де $T_{\text{ном}}(t)$ - температура в приміщенні в момент часу t , $V_{\text{ном}}$ - об'єм приміщення, $v_n(t)$ - швидкість потоку повітря, що виходить з кондиціонера в момент t , $\sigma_{\text{конд}}$ - пропускний перетин кондиціонера, $T_{\text{ном}}(t)$ - температура в приміщенні в момент часу t , $T_n(t)$ - температура потоку повітря в момент часу t , $P_{\text{дв}}$ - потужність двигуна кондиціонера, $d_i(t)$ - прийняте в момент t рішення про зміну температури в приміщенні з поточної температури $T_{\text{ном}}(t)$ до температури T , $\varphi_1(d_i(t), P_{\text{дв}})$ - функція, що описує залежність швидкості $v_n(t)$ подається в приміщення повітря від потужності двигуна кондиціонера $P_{\text{дв}}$ і T , $\varphi_2(d_i(t), P_{\text{дв}})$ - функція, що описує залежність температури $T_n(t)$ подається в приміщення повітря від потужності двигуна кондиціонера $P_{\text{дв}}$ і T . Структура функцій $\varphi_1(d_i(t), P_{\text{дв}})$ і $\varphi_2(d_i(t), P_{\text{дв}})$ - визначається виключно технічними характеристиками кондиціонера. Однак, безпосередньо зміна швидкості подачі повітря в приміщення і температури подачі зазвичай має комутаційний тип, тобто змінюється стрибкоподібно, що також дозволяє зробити висновок про відсутність гладких властивостей у функції $g(\cdot)$.

Виходячи з вищесказаного, для вирішення завдання (2.1) - (2.4) пропонується використовувати методи нульового порядку, в яких не робиться ніяких припущень про властивості функції $f()$. Найбільш відомим з подібних методів є метод покоординатного спуску. Наведемо опис цього методу, адаптоване до вирішення даної задачі. Обмежимося для простоти опису тільки одним основним ресурсом стосовно цільової функції Z_1 .

На вході алгоритму задана цільова функція

$$Z_1 = Z(\{s_i^j(t)\}; i \in D_v, j = \overline{1; K_i}) = \frac{1}{T} \sum_{t=1}^T \sum_{i \in D_v} \sum_{j=1}^{K_i} \omega_{ij} (s_{i,pez}^j - s_i^j(t))^2, \quad (2.6)$$

обмеження (2), (3), (4) і початковий стан системи, що задається набором показників $M_0 = (\{S_i^j(0)\}; i \in D_v, j = \overline{1; K_i})$. Тоді суть алгоритму полягає в побудові послідовності точок M_1, M_2, \dots, M_n до тих пір, поки не будуть виконані умови критерію зупинки. Наведемо послідовні кроки методу.

1. Нехай точки M_1, M_2, \dots, M_L вже побудовані. Послідовно перебираються в лексикографічному (зростаючий) порядку пари $(i; j)$, $i \in D_v, j$ від 1 до K_i . Тобто при фіксованому i послідовно збільшується значення j до значення K_i , після чого значення i збільшуємо на одиницю і вважаємо $j = 1$. Для кожної пари $i = I$ і $j = J$ виконуємо наступні дії.

2. Фіксуємо значення всіх змінних S_i^j , крім $i = I$ і $j = J$, на рівні їх значень в поточній точці M_L ; тобто, якщо $M_L = \{S_i^j(L)\}$, то вважаємо $S_i^j = S_i^j(L)$ для всіх $i \in D_v; j = \overline{1; K_i}$. Отримуємо одновимірну функцію $Z_1(S_I^J)$, з відповідними обмеженнями (2.2), (2.3).

3. Проводиться поліпшення значення цільової функції по змінно $(I; J)$ наступним чином. Побудуємо дві допоміжні точки $M_{L+1}^{\wedge}(I; J)$ і $M_{L+1}^{\ddot{}}(I; J)$: для точки $M_{L+1}^{\wedge}(I; J)$ координати $S_i^j = S_i^j(M_{тек})$ для всіх $(i; j)$ таких, що $i \neq I$ і $j \neq J$; $S_I^J = S_I^J(M_{тек}) - \delta_J$, де δ_J - допустима точність аргументу оптимальної точки по j -ому показнику основного ресурсу. Аналогічно знаходяться координати допоміжної точки $M_{L+1}^{\ddot{}}(I; J)$: $S_i^j = S_i^j(M_{тек})$ для всіх $(i; j)$ таких, що $i \neq I$ і $j \neq J$; $S_I^J =$

$S_I^J(M_{тек}) + \delta_J$. Перевіряємо виконання умов (2.2), (2.3) і (2.4) в точках $M_{L+1}^{\dot{}}(I; J)$ і $M_{L+1}^{\ddot{}}(I; J)$ при деяких рішеннях $d_k = d_k(I; J) \in D$ (для кожної з точок $M_{L+1}^{\dot{}}(I; J)$ і $M_{L+1}^{\ddot{}}(I; J)$ свій варіант вирішення; відповідне d_k знаходиться шляхом перебору всіх рішень їх D). Точки, в яких ці умови не виконуються, викидаються і в подальшому не розглядаються. Нехай обидві точки задовольняють обмеженням (2.2), (2.3) і (2.4).

Порівнюються значення $Z_1(M_{тек})$, $Z_1(M_{L+1}^{\dot{}}(I; J))$ і $Z_1(M_{L+1}^{\ddot{}}(I; J))$: якщо $Z_1(M_{L+1}^{\dot{}}(I; J)) < Z_1(M_{тек})$ і $Z_1(M_{L+1}^{\dot{}}(I; J)) < Z_1(M_{L+1}^{\ddot{}}(I; J))$, то вважаємо $M_{тек} = M_{L+1}^{\dot{}}(I; J)$; $Z_1(M_{L+1}^{\ddot{}}(I; J))$ якщо $Z_1(M_{L+1}^{\ddot{}}(I; J)) < Z_1(M_{тек})$ і $Z_1(M_{L+1}^{\ddot{}}(I; J)) < Z_1(M_{L+1}^{\dot{}}(I; J))$, то вважаємо $M_{тек} = M_{L+1}^{\ddot{}}(I; J)$; в інших випадках точка $M_{тек}$ не змінюється. Якщо тільки одна з точок $M_{L+1}^{\dot{}}(I; J)$ і $M_{L+1}^{\ddot{}}(I; J)$ задовольняє умовам (2.2) і (2.3), то останні умови перевіряються тільки для цієї точки. Якщо жодна з точок $M_{L+1}^{\dot{}}(I; J)$ і $M_{L+1}^{\ddot{}}(I; J)$ не задовольняє умовам (2.2) і (2.3), то переходимо до наступного кроку алгоритму.

4. Якщо поточна пара індексів $(I; J) \neq (I_{max}; K_{I_{max}})$ (тобто ми ще не дійшли до останньої пари індексів $(i; j)$), то переходимо до наступної по лексикографічному порядку пари, тобто вважаємо $I := I + 1$, $J := 1$, якщо $J = K_i$, і $J := J + 1$ (I не змінється), якщо $J < K_i$. Тут $I_{max} = |D_v|$ і загальне число датчиків, пов'язаних з v -им ресурсом. Потім переходимо до кроку 2.

5. Якщо поточна пара індексів $(I; J) \neq (I_{max}; K_{I_{max}})$ (тобто всі пари $(i; j)$) переглянуті, то вважаємо $M_L = M_{тек}$.

6. Перевіряємо критерій закінчення пошуку: $|Z_1(M_{L+1}) - Z_1(M_L)| < \varepsilon$, де ε - необхідна точність кінцевого результату. Якщо остання нерівність виконується, то процес пошуку закінчується; рішенням є набір рішень $\{d_k(I; J)\}$, Отриманих при побудові точки M_L . Ці рішення повинні послідовно реалізовуватися відповідно до послідовності їх побудови. Якщо остання нерівність не виконується, то переходимо до побудови наступної точки M_{L+1} (крок 1).

Опис алгоритму закінчено. Відзначимо, що в процесі перебору варіантів рішень d_k на кроці 3 для підвищення швидкості перебору можна скористатися

введеної вище впорядкованістю наборів рішень d_k і координат S_i^j , оскільки зазвичай компоненти функція $g()$ в умови (2.2) монотонні за рішеннями d_k і координатами S_i^j . Наприклад, в підсистемі ОВК чим вище необхідна температура, швидкість повітря, вміст вуглекислого газу або вологості в повітрі, тим більше значення показників S_i^j датчиків.

2.3 Аналіз факторів впливу на функціонування системи «Розумний будинок»

Постає завдання формування системи збору даних. Для цього, перш за все, необхідно сформувавши склад тих даних, збір яких необхідний для ефективної роботи СУРБ.

Для цього спочатку проводимо системну класифікацію всіх факторів, які можуть вплинути на керування будівлею або суб'єктів, що перебувають у ньому на законних підставах - назовемо зазначену сукупність об'єктом управління. Потім на основі результатів класифікації факторів сформований відносно повний склад необхідних датчиків.

На основі аналізу складу всіх можливих типів впливів виділені групи чинників, пов'язані з наступними компонентами об'єкта управління і СУРБ, показаними на рис. 2.2 [9]:

1. Безпосередньо будівельні конструкції і їх окремі елементи, що формують сам будинок;
2. Інфраструктурні системи і їх елементи, пов'язані з системами життєзабезпечення людей (енерго-, тепло-, водо- та газопостачання, освітлення), а також з наданням типових послуг населенню (зв'язок, інтернет, вентиляція і кондиціонування);
3. Інфраструктурні системи, що підтримують будівлю в стані, що задовольняє всім нормативним вимогам, а також забезпечують контроль за станом усіх елементів, систем, конструкцій будинку і управління будинком;

4. Технічні та програмно-апаратні компоненти, що входять до складу СУРБ;
5. Суб'єкти, що знаходяться на площах, в приміщеннях і території інтелектуальної будівлі на законних або незаконних підставах;
6. Системи охорони і безпеки інтелектуальної будівлі;
7. Різні дестабілізуючі зовнішні і внутрішні впливи природного, техногенного та суб'єктивного характеру, які можуть порушити нормальний режим роботи інтелектуальної будівлі, його підсистем і навіть зруйнувати частина з них;
8. Безпосередньо СУРБ як частина об'єкта управління.

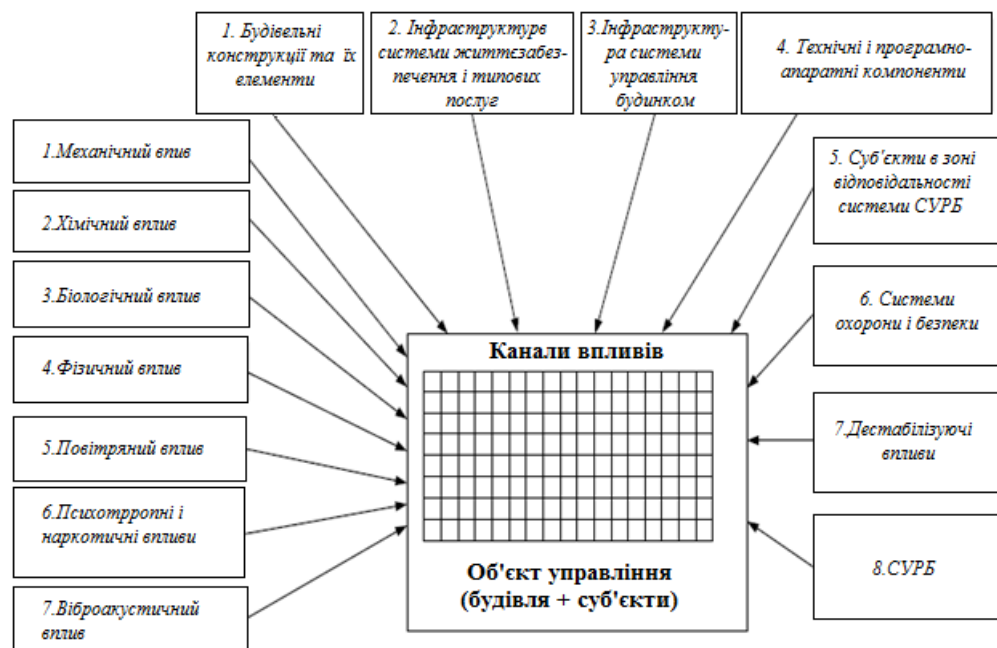


Рисунок 2.2 - Діаграма каналів впливу на об'єкт управління

Деталізація перерахованих факторів на наступному рівні, що враховує змістовний склад кожного з цих компонентів, дозволила виділити більше 56 можливих джерела впливу на об'єкт управління.

Будівля і суб'єкти, що знаходяться в ньому, піддаються впливу різних чинників, як внутрішніх, так зовнішніх. Аналіз можливих способів впливу можна виділити наступні класи способів впливу:

1. Механічні дії - обвалення або руйнування елементів і конструкцій будівель, і приміщень, падіння інших предметів, що знаходяться в зоні

переміщення і знаходження суб'єктів, слизькі поверхні (зокрема, ожеледиця), падіння предметів з висоти (зокрема снігу з даху, бурульок), суспензії, пил, в тому числі що містять не виводяться і канцерогенні добавки;

2. Хімічні впливи - речовини, небезпечні для здоров'я людей або для працездатності персоналу, вибухо- і пожежонебезпечні, сильно і неприємно пахнуть, небезпечні для технічних пристроїв, алергени;

3. Біологічні впливу - мікроорганізми (бактерії, віруси та ін.) І макроорганізми (рослини і тварини);

4. Фізичні впливу (немеханічної природи) - впливу у вигляді фізичних полів і випромінювань: електромагнітних полів, рентгенівських, радіоактивних, іонізуючих і ультрафіолетових випромінювань і ін .;

5. Повітряні впливу -димові, пилові та інші завіси, витоку газу, сильні вихрові і турбулентні руху;

6. Психотропні та наркотичні впливу;

7. Віброакустичного впливу - сильні звукові сигнали, вібраційні впливу механічного характеру.

2.4 Підсистема збору вхідних даних



Рисунок 2.3 - Діаграма вхідних даних систем управління будівлею

Процес збору та накопичення даних можна розбити на ряд відносно незалежних етапів:

а) безпосередньо збір даних за допомогою різних технічних, програмно-апаратних засобів, а також з суб'єктних джерел;

б) реєстрація даних на носіях інформації;

в) аналіз даних по критеріях повноти, адекватності, несуперечності, актуальності, своєчасності, безпеки, довіри і ін .;

г) прийняття адекватних рішень при виявленні порушень хоча б по одному з перерахованих критеріїв і реалізації прийнятих рішень;

д) передача отриманих та проаналізованих даних в центральних блок управління і відповідні функціональні підсистеми для виконання своїх обов'язків.

З перерахованих етапів обробки даних одним з найбільш важливих є етап безпосередньо збору даних. Загальна схема формування потоку вхідних даних приведена на рис. 2.3.

На зазначеній діаграмі виділено сім можливих джерел вхідних даних:

1. Датчики і датчикові пристрої, що є комплекси, що включають більше одного датчика. На основі датчиків здійснюється збір даних по всіх технічних елементів системи життєзабезпечення будинку (зокрема, водо- і енергопостачання, комунальні послуги), по багатьом елементам системи безпеки. Даний джерело даних є основним для забезпечення ефективного функціонування СУРБ.

2. Система відеоспостереження, що є одним з основних джерел інформації для служби безпеки. Ця система дозволяє також вести моніторинг за багатьма елементами системи життєзабезпечення інтелектуального будинку, стан яких може бути ідентифіковано на основі візуального спостереження.

3. Інтернет як джерело інформації, що дозволяє, перш за все, допомогти в знаходженні можливих варіантів вирішення різних проблем, а також в отриманні інформації про різні фізичних осіб, які перебувають або потрапили в сферу професійних завдань підсистеми безпеки СУРБ.

4. Журнал подій, де повинні фіксуватися всі значущі події в зоні контролю СУРБ, дозволяє на основі його аналізу виявити можливі події, проблеми,

неприємності, загрози, тенденції, а також спрогнозувати можливі стану всієї системи, що включає саму будівлю з усією зоною контролю і СУРБ.

5. Інформація від ЛЗ, що знаходяться в зоні контролю СУРБ на законних підставах - мешканці будинку, відвідувачі, орендарі, співробітники і т.п.), яка може бути в тому числі і за допомогою опитування з боку персоналу СУРБ (керівництва СУРБ, служб безпеки і життєзабезпечення). Ця інформація може сприяти більш адекватної та оперативній роботі СУРБ.

6. Інформація від персоналу СУРБ, в тому числі і суб'єктивного характеру, що стосується всіх аспектів функціонування СУРБ. Дана інформація дозволяє виявити потенційні і реальні проблеми, які поки знаходяться поза фіксації за допомогою інших джерел даних.

7. Законодавчі і нормативні документи, що регламентують діяльність як СУРБ в цілому, так і роботу окремих її підсистем, включаючи підсистему збору і підготовки даних. Зазвичай виділяють в ній дві групи: перша група - законодавчі та нормативні вимоги та обмеження різних державних регуляторів (зокрема, санітарно-епідеміологічної служби, пожежної та екологічної безпеки, міністерства внутрішніх справ та ін. І друга група - рішення і нормативні вимоги та обмеження керуючого органу будівлі.

Одна з основних функцій підсистеми збору та підготовки даних є забезпечення безперебійного, надійного і адекватного функціонування всіх перерахованих вище джерел даних [23]. У цьому полягає регулятивна функція підсистеми збору та підготовки даних. Однак, обмеження завдань підсистеми збору та підготовки даних лише завданнями регулятивного характеру може виявитися недостатнім в умовах виникнення нестандартних, і особливо, аварійних і АС, оскільки в подібних випадках можуть знадобитися більш адаптивні дії підсистеми збору та підготовки даних; зокрема, більш частий збір даних (наприклад, при виникненні пожежі), пошук додаткової інформації з інших (зовнішніх) джерел (наприклад, при виникненні небезпеки можливих протиправних дій). Тому на систему збору та підготовки даних покладаються додаткові функції, пов'язані з

адаптивними змінами в процесі збору та підготовки даних, що додає до регулятивних функцій підсистеми також елементи управління.

На схемі рис. 2.3 наведені також можливі дії, що управляють в рамках системи збору та підготовки даних, саме:

а) управління частотою збору і фіксації даних, а також їх попередньої обробки з урахуванням умов функціонування окремих датчиків (наприклад, при підвищеній температурі, високій вологості);

б) пошук і активізації додаткових джерел опитування (суб'єктів, інформаційних систем);

в) пошук відповідей і рішень з виниклих проблем в глобальних мережах.

Відзначимо, що теоретично дані від кожного датчика можуть збиратися і накопичуватися безперервним чином. Однак, якщо ситуація відносно стабільна, незмінна, то немає необхідності збирати дані в безперервному режимі, тобто витрачається великий обсяг обчислювальних ресурсів (включаючи пам'ять), які в своїй основній масі не затребувані. Таким чином, при стабільній ситуації дані доцільно збирати досить рідко. Навпаки, якщо ситуація динамічно і швидко змінюється, то дані необхідно збирати дуже часто для того, щоб безперервно мати інформацію про поточний стан ситуації. Таким чином, є оптимальна частота збору даних по кожній ситуації, при якій, з одного боку, дані ненадлишкових, а, з іншого боку, їх досить для ефективного контролю ситуації. Тому, доцільно змінювати частоту збору даних в залежності від динамічності зміни і важливості конкретного процесу або події.

Основний («центральный») блок «Система управління процесом збору та підготовки вихідних даних». Опишемо технологію його функціонування рис. 2.4. Оскільки розглянута система повинна реалізувати також і функції управління процесом збору вихідних даних, то необхідно, щоб вико-мі датчики були здатні реагувати на дії, тобто як мінімум вони повинні бути адресними. Більш того, пропонується (в міру наявних фінансових можливостей) використовувати «інтелектуальні датчики», які мають великий набір керованих параметрів. Для підвищення ефективності управління процесом збору даних доцільно розбити всі

датчики на групи (за територіальним розташуванням, за функціональним призначенням і т.п.) і датчики кожної з груп підключити до мікропроцесора. Він повинен здійснювати окремі функції контролю і управління для відповідної групи датчиків - зокрема, контроль за станом датчиків (по всій сукупності експлуатаційних і призначених для користувача характеристик). Крім того, при виникненні відхилень датчиків від їх регламентних станів програмне забезпечення мікропроцесора має негайно інформувати про це диспетчерський центр СУРБ. Крім цього, управлінський вплив з центру управління СУРБ стосовно конкретного датчику або сукупності датчиків також доцільно передавати через МК - оскільки від них в центрі управління може бути отримана повна інформація по доставці і фактичній реалізації керуючого впливу. Відзначимо, що для підвищення швидкості обміну, все МК передають дані через загальну шину.

Процес збору та обробки даних здійснюється безперервно (циклічно) і може бути частково або повністю зупинений тільки при безпосередньо керуючого дії з центру управління СУРБ або при фактичному руйнуванні СУРБ - це відображено в блоках 21, 22, 23 рис. 2.4.

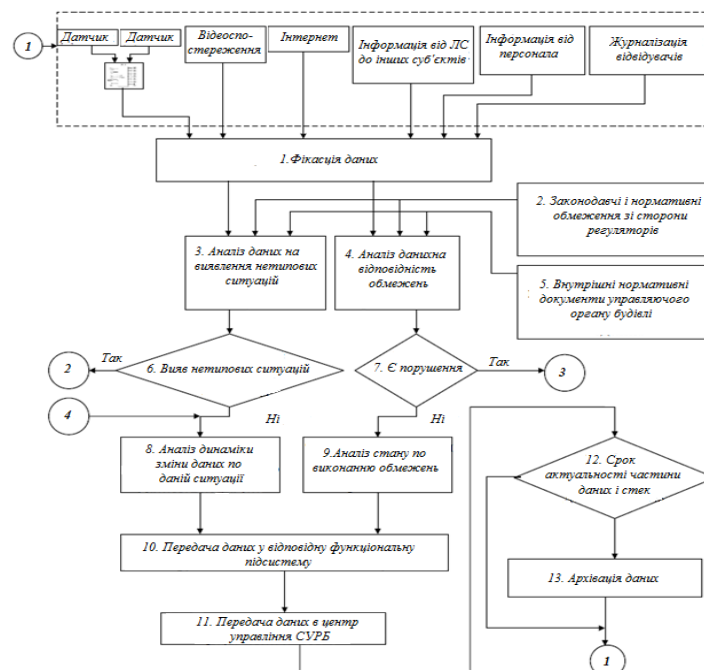


Рисунок 2.4 - Загальна схема функціонування підсистеми управління збором і протоколами (МК - мікроконтролери)

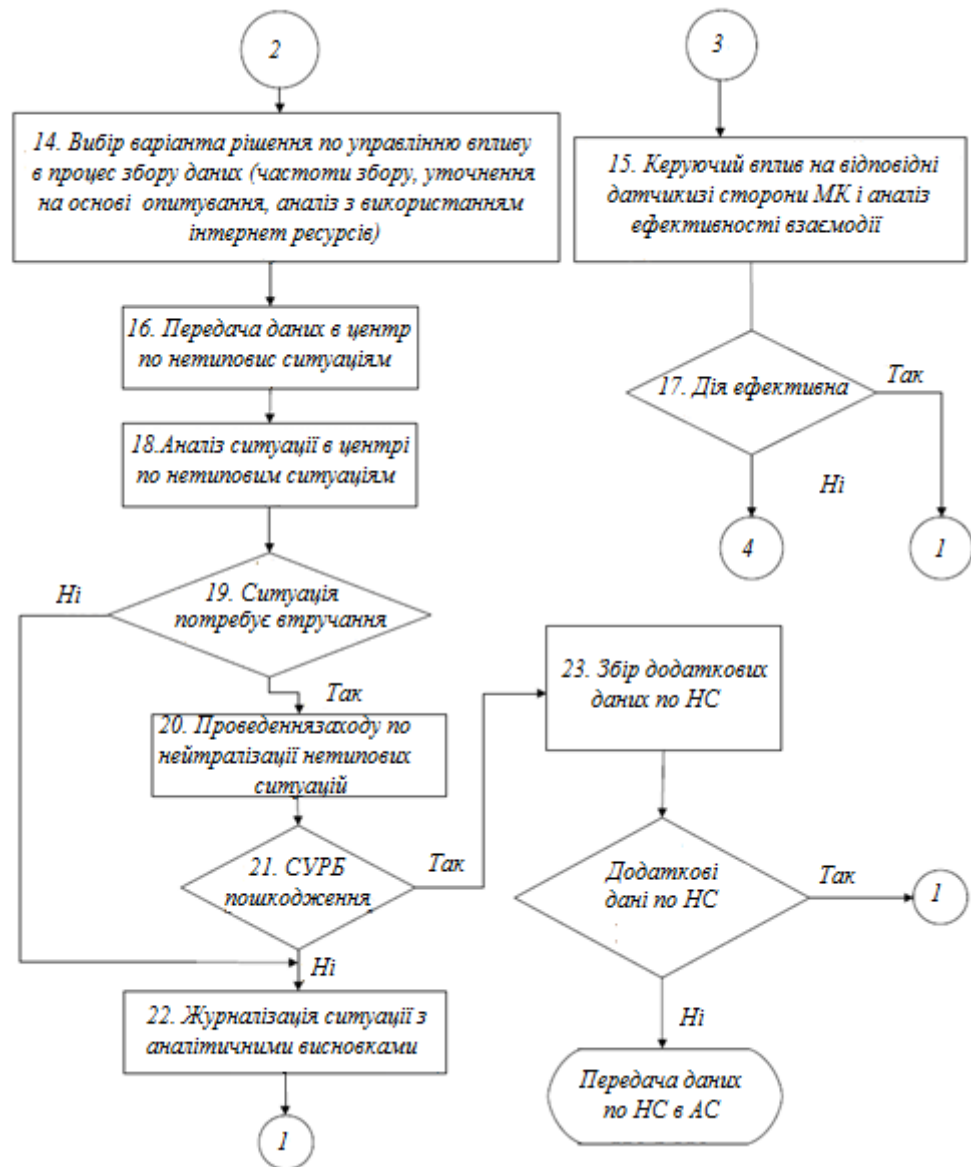


Рисунок 2.5 - Загальна схема функціонування підсистеми контролю збором і протоколами (продовження)

Безпосередньо процес збору даних з первинних джерел, які перераховані вище, представлений на рис. 2.4 блоками 0, 2, 5. Потім відбувається реєстрація даних (блок 1 на рис. 2.4), де всі отримані дані в відпо-відно до прийнятими для кожного їх типу форматами розміщуються в оперативній пам'яті. Далі в ній відбувається аналіз даних за двома групами вимог (блоки 3, 4): (А) на відповідність усім вимогам і обмеженням, які диктуються нормативними документами різних рівнів, паспортними обмеженнями відповідних датчиків, або знаходяться в

суперечності з іншими отриманими даними; (Б) на наявність ознак нетипової ситуації в тому числі предаварійною або аварійною.

Якщо зазначених проблем немає, то обробка даних продовжується в відповідно до прийнятої технологією (блоки 8-13): аналіз зміни даних, що надходять і ситуації в часі (блоки 8, 9); передача даних до відповідних функціональних блоків (блок 10) і в центр управління СУРБ (блок 11); де вони розміщуються для регламентного накопичення і зберігання, перевірки даних, наявних в накопичувачах, на предмет їх старіння. Якщо дані перестали бути актуальними (блок 12), то вони передаються в «блок архівації» для довготривалого зберігання. Потім процедура збору і реєстрації даних повертається до початку – для реалізації наступного циклу своєї роботи.

У разі, якщо порушується хоча б одне із зазначених вище двох груп умов (наприклад, ситуація стає нетиповою), то система реалізує певний набір дій, що управляють (блоки 14 - 23): вибираються управляючі впливи на відповідні датчики (блоки 14, 16, 18) , в тому числі що призводять до зміни частоти збору даних; здійснюється пошук по зовнішніх джерел даних; проводиться пошук рішень в аналогічних ситуаціях. Якщо ситуацію не вдається нормалізувати в прийнятні терміни (блок 19), то вся інформація по ній передається в «центр управління», в якому передбачена спеціальна служба по діям в нетипових ситуаціях (блок 20). Ця служба вибирає варіанти рішень; керує їх реалізацією; при необхідності сповіщає всіх суб'єктів про виниклу ситуацію.

Якщо і ця служба виявилася нездатною виправити ситуацію, то виникає небезпека можливого пошкодження і навіть руйнування СУРБ або її частини (блоки 21, 23, 24). У свою чергу це може призвести до повної або часткової блокування роботи підсистеми збору та підготовки даних.

При порушенні регламентних обмежень за даними система управління, перш за все, намагається шляхом діагностування відповідних датчиків виявити можливе порушення адекватної роботи окремих з них (блоки 15), зокрема, можлива відмова або вихід з ладу цих датчиків. Відповідно здійснюється прийняття рішень щодо їх подальшої експлуатації з подальшою реалізацією. Якщо ж проведені заходи не

привели до відновлення адекватної роботи датчиків (блок 17), то ситуація переводиться в категорію нетипових відхилень і обробляється як нетипова ситуація по описаній вище технології.

Розглянемо приклад, що управляють, з боку підсистеми збору та підготовки даних. Нехай нетиповою ситуацією є виникнення пожежі (загоряння) в будівлі. У цьому випадку дані від сповіщувачів пожежної сигналізації надходять в блок 1 і потім відповідно до описаної технологією, доходять до блоків 3, 14. З них дані (сигнал) передаються в центр управління СУРБ. При необхідності (в залежності від масштабів загоряння) формуються команди на включення світлових, звукових і мовних сповіщувачів в зоні спалаху. Безпосередньо підсистема збору і підготовки даних паралельно формує керуючі сигнали для виконання іншими системами, що входять в неї, за наступним алгоритмом.

1. Система телевізійного спостереження «направляє» найближчі відеокамери на вогнище спалаху і привласнює цих відеокамер вищий пріоритет, забезпечуючи тим самим безперервне надходження з них відеоданих із зони загоряння. На моніторі спостереження формуються укрупнені зображення з цих відеокамер. Система відеофіксації переходить в режим пріоритетною записи зображень із зони загоряння.

2. Система управління мікрокліматом вимикає приточну систему вентиляції, яка обслуговує дану зону - щоб запобігти надходженню свіжого повітря (що містить кисень) до вогнища спалаху. Для видалення диму з коридорів, холів, сходів (уздовж маршрутів евакуації) включається відповідна підсистема димовидалення (відкриваються заслінки, включаються вентилятори, які обслуговують витяжну систему).

3. Чи включається система автоматичного пожежогасіння - якщо окремі приміщення або будівля в цілому обладнано такою системою.

4. Система управління електропостачанням відключає ланцюга електроживлення від обладнання і освітлювальних пристроїв в зоні пожежі.

5. Система управління освітленням включає аварійне освітлення.

6. Система управління доступом розблокує двері (наприклад, обладнаний-ні кодовими замками) для безперешкодної евакуації людей.

7. Система голосового оповіщення включає голосові оповіщення у відповідних частинах будівлі.

8. Система управління ліфтами спустить на перший поверх і при необхідності відключити їх і т.д.

Одночасно сигнал про наявність вогнища загоряння надходить в єдиний диспетчерський центр СУРБ на автоматизоване робоче місце оператора. На моніторі АРМ з'являється графічна і текстова інформація про пожежу і місце його виникнення. Оператор має можливість проконтролювати роботу системи автоматики, при необхідності продублювати її, підключити (викликати) пожежні підрозділи МНС для боротьби з пожежею.

Описана технологія роботи підсистеми збору та підготовки даних дозволяє на більш якісному рівні вирішувати багато важливих завдань, пов'язані з функціонуванням будівлі і забезпеченням комфортності і безпеки перебування всіх ЛЗ в ньому [11]. Практична реалізація наведеної технології, як це видно з наведеного прикладу, вимагає подальшої деталізації цієї схеми. За відносно типових ситуацій, в тому числі і аварійного характеру (напри-заходів, розкрадання або пошкодження окремих датчиків), підсистема збору і підготовки даних може «вирішувати питання» в рамках своїх повноважень. Однак, як було показано вище, при серйозних відхиленнь у показаннях датчиків від нор-мінімальних значень, рішення повинно прийматися в центрі СУРБ. Прикладами таких ситуацій можуть бути пожежі, повені, руйнування несучих конструкцій будівлі або фундаменту, вихід з ладу систем опалення в період зимових холодів, масові заворушення та ін.

Таким чином, сформована загальна технологія функціонування підсистеми збору та підготовки даних, яка є однією з необхідних для забезпечення функціонування СУРБ і її функціональних підсистем.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ «РОЗУМНОГО БУДИНОКУ»

3.1 Алгоритм функціонування системи управління безпекою будинку

В системі контролю безпеки «Розумного будинку» моніторинг параметрів здійснюється на підставі показників датчиків. Датчики встановлюються в приміщенні. Інформація з датчиків надходить на локальні контролери, які її обробляють і на основі отриманих даних здійснюють регулюючий вплив. Центральний контролер збирає інформацію з локальних контролерів, за запитом передає її на 1 рівень, приймає і виконує призначені для користувача команди, задає параметри локальним контролерам.

Призначена для користувача система контролю - це різноманітні мобільні і стаціонарні пристрої, за допомогою яких можливо реалізувати безпеку будинку. Це може бути смартфон або персональний комп'ютер. Для віддаленого доступу до підсистеми потрібно підключитися до мережі і ввести пароль. Крім того, можливо здійснювати деякі дії, що управляють за допомогою відправки SMS зі свого мобільного телефону.

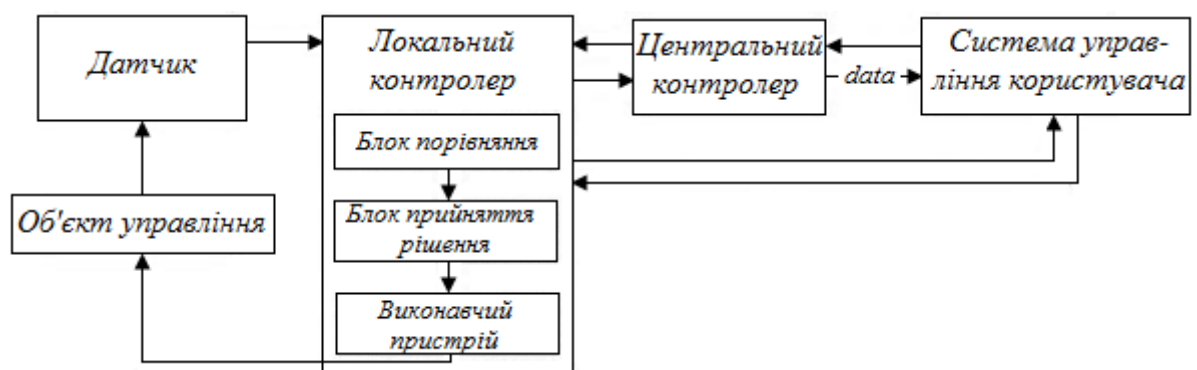


Рисунок 3.1 - Алгоритм роботи системи контролю

Інформація про стан об'єкта управління фіксується датчиком і передається на локальний контролер через певні проміжки часу або по необхідності (наприклад, аварійне спрацювання датчика вторгнення). Центральний контролер задає

еталонне значення регульованого параметра локальному контролеру. Локальний контролер функціонально складається з блоку порівняння, блоку прийняття рішення і виконавчого пристрою. У блоці порівняння знаходиться різниця чинного і еталонного значення регульованого параметра.

На підставі величини і знака цієї різниці у відповідному блоці приймається рішення про необхідність регулюючого впливу, яке здійснюється виконавчим пристроєм. Центральний контролер збирає з локальних контролерів дані про стан об'єктів управління, задає локальним контролерам еталонні значення, передає на 1 рівень запитувані параметри, приймає і виконує призначені для користувача команди. Як видно з функціональної схеми, користувачі безпосередньо можуть задавати параметри локальним контролерам і отримувати від них необхідну інформацію.

3.2 Вибір апаратних засобів та елементів системи

В якості центрального контролера був обраний Arduino, так як це недорогий мікропроцесор з відкритою архітектурою, здатний зчитувати вхідні дані у вигляді напруги на своїх аналогових контактах. Якщо до певних входів пристрою підключити датчики, то воно буде програмним способом зчитувати інформацію з цих контактів.

Arduino – це платформа для розробки електронних пристроїв з автоматичним, напіваавтоматичним або ручним керуванням. Вона виконана за принципом конструктора з чітко визначеними правилами взаємодії між елементами. Система є відкритою, що дозволяє стороннім виробникам брати участь в її розвитку. Платформа Arduino гарна саме тим, що не замикається на певному виробнику, а дозволяє самому підібрати відповідні компоненти. Їх вибір величезний, тому можна реалізувати практично будь-які задуми.

Крім різноманіття пристроїв, що підключаються, варіативності додає середовище програмування, реалізоване на мові C++. Користувач може не тільки

скористатися створеними бібліотеками, а й сам запрограмувати реакцію компонентів системи на виникаючі події [24].

Материнська плата об'єднує в собі такі елементи:

- мікроконтролер (процесор). Основне його призначення - видавати і вимірювати напругу в портах в діапазоні 0-5 або 0-3.3 В, запам'ятовувати дані і робити обчислення;

- програматор. За допомогою цього пристрою в пам'ять мікроконтролера записують програму, згідно з якою буде працювати "розумний будинок". До комп'ютера, планшета, смартфона або іншого пристрою його підключають за допомогою USB-інтерфейсу;

- стабілізатор напруги. Необхідно пристрій на 5 вольт, потрібно для живлення всієї системи.

На материнській платі мікроконтролера існують два види портів:

- цифрові, які позначені на платі буквами "d";

- аналогові, які позначені літерою "a".

Завдяки їм мікроконтролер здійснює зв'язок з підключеними пристроями.

Будь-порт може працювати як на отримання сигналу, так і на його віддачу.

Цифрові порти з позначкою "rwm" призначені для введення і виведення сигналу типу ШИМ (широко-імпульсна модуляція). Кожен мікропроцесор оснащений трьома видами пам'яті:

- flash Memory. Основна пам'ять, де зберігається код програми управління системою. Незначну її частину (3-12%) займає вшита програма завантаження (bootloader);

- sRAM. Оперативна пам'ять, де зберігаються тимчасові дані, необхідні при роботі програми. Відрізняється високою швидкістю роботи;

- EEPROM. Більш повільна пам'ять, де також можна зберігати дані.

Основна відмінність видів пам'яті для зберігання даних полягає в тому, що при виключенні електроенергії інформація, яка записана в SRAM, втрачається, а в EEPROM залишається. Але у енергонезалежного типу є і недолік – обмежене число циклів запису. Це потрібно пам'ятати при створенні власних додатків [24].

1). Всі модулі, які будуть підключені до плати, мають як мінімум три виходи. Два з них – дроти живлення, а також напруга 5 або 3.3 В. Третій провід є логічним. По ньому йде передача даних до порту. Для підключення модулів використовують спеціальні згруповані по 3 штуки дроти, які іноді називають «джамперами».

Для збільшення можливостей материнських плат використовують Шилди (Shields). Ці додаткові пристрої розширюють функціонал основної плати. Вони виготовляються під конкретний форм-фактор, що відрізняє їх від модулів, які підключають до портів. Шилд коштує дорожче модулів, однак робота з ним простіше. Також вони забезпечені готовими бібліотеками з кодом, що прискорює розробку власних програм управління для "розумного будинку". Підключення модулів від сторонніх виробників і можливість роботи з ними, використовуючи вбудовану мову програмування – основна перевага відкритої системи Arduino в порівнянні з "фірмовими" рішеннями для "розумного будинку". Головне, щоб модулі мали опис одержуваних або переданих сигналів. Способи отримання інформації: введення інформації може бути здійснений через цифровий та аналоговий порти. Це залежить від типу кнопки або датчика, який отримує інформацію і транслює її на плату. Сигнал до мікропроцесора може бути посланий людиною, який використовує для цього два способи: натискання кнопки (клавіші). Логічний провід в цьому випадку йде до цифрового порту, які отримує значення "0" в разі відпущеної кнопки і "1" в разі її натискання.

Обертання ковпачка поворотного потенціометра (резистора) або зрушення важеля движкового. В цьому випадку логічний провід йде до аналогового порту. Напруга проходить через аналогово-цифровий перетворювач, після чого дані надходять до мікропроцесора. Кнопки використовують для старту якої-небудь події, наприклад, включення і виключення світла, опалення або вентиляції. Поворотні ручки застосовують для зміни інтенсивності – збільшення або зменшення яскравості світла, гучності звуку або швидкості обертання лопатей вентилятора.

Крім збору та аналізу інформації "розумний будинок" повинен реагувати на виникаючі події. Присутність на сучасних побутових приладах просунутої

електроніки дозволяє звертатися до них безпосередньо, використовуючи Wi-Fi, GPRS/EDGE або Ethernet. Зазвичай, для систем Arduino реалізують комутацію мікропроцесора і високотехнологічних пристроїв за допомогою Wi-Fi. Крім "спілкування" з комп'ютеризованими приладами часто виникають завдання, пов'язані з виконанням будь-яких механічних дій. Наприклад, до плати можна підключити сервопривід або невеликий редуктор, який буде від неї живиться. У разі необхідності підключення потужних пристроїв, що працюють від зовнішнього джерела живлення, використовують два варіанти:

- включення в електричний ланцюг електромагнітного або твердотільного реле. Це пристрій замикає й розмикає один з проводів по команді, що надходить від мікропроцесора. Основна їх характеристика – максимально допустима сила струму (наприклад, 40 А), яка може проходити через цей прилад;

- підключення силового ключа – мосфети для постійного струму і симистора для змінного. Вони володіють меншим значенням допустимої сили струму (5-15 А), але можуть плавно збільшувати навантаження. Саме для цього на платах передбачені ШІМ-порти. Цю властивість використовують при регулюванні яскравості освітлення, швидкості обертання вентиляторів і т.д.

Комплектуючі. Arduino Piranha UNO – ця плата є повним аналогом Arduino UNO R3, але на відміну від Arduino UNO R3 на платі Piranha UNO R3 використовуються більш потужні стабілізатори напруги для шин живлення на 5В і 3.3В, що призводить до більш стабільної роботи більшості модулів рис.3.2.



Рисунок 3.2 – Монтажна реалізація елементної бази плати Arduino Piranha

Uno

Приклад розпіновки плати представлено на рис.3.3.

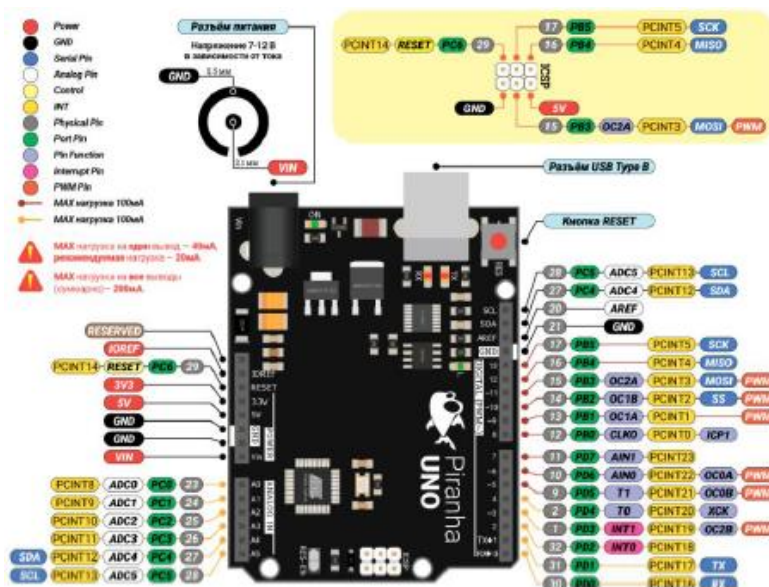


Рисунок 3.3 – Розпіновка плати Arduino

GSM/GPRS Shield A6 – дозволяє працювати в мережах стільникового зв'язку за технологіями GSM/UMTS/LTE для прийому/передачі даних, відправлення/отримання SMS і здійснення телефонного голосового зв'язку рис.3.4.



Рисунок 3.4 – Монтажна реалізація елементної бази плати GSM/UMTS/LTE Shield A6

Battery Shield встановлюється на Arduino, забезпечуючи її живленням як звичайний акумулятор забезпечує живленням смартфон, планшет і т.д.



Рисунок 3.5 – Монтажна реалізація Battery Shield 2

Якщо до шин живлення Arduino підключені інші пристрої, вони також отримують живлення від Battery Shield рис.3.5.

Це плата розширення, яка спрощує процес підключення модулів до Arduino. Використання Trema Shield рис.3.6 позбавляє від необхідності пайки проводів при підключенні декількох модулів до Arduino, спрощує процес створення пристроїв.

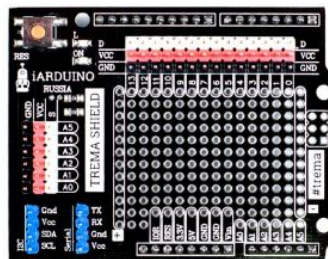


Рисунок 3.6 – Монтажна реалізація Trema Shield

Trema-модуль цифровий термометр: вимірює температуру навколишнього середовища. Цифровий сигнал стійкіший від наведень, на відміну від аналогового датчика температури рис.3.7.



Рисунок 3.7 – Монтажна реалізація елементної бази цифрового термометру



Рисунок 3.8 – Цифровий термометр DS18B20

Цифровий термометр (герметичний): датчик температури DS18B20 – дозволить вимірювати температуру навколишнього середовища рис. 3.8.

Трема-модуль – датчик газу MQ-2 (широкого спектру): даний модуль здатний визначати концентрацію широкого спектра газів в повітрі (природні гази, вуглекислий і чадний газ, вуглеводні, дим) рис.3.9.



Рисунок 3.9 – Монтажна реалізація елементної бази датчик газу MQ-2

ИК-датчик руху HC-SR501: він дозволяє виявити рух людей в контрольованій зоні за рахунок випромінюваного людиною тепла рис.3.10.



Рисунок 3.10 – ІК-датчик руху HC-SR501

Магнітний сповіщувач (геркон): це перемикач і магніт, поміщені в пластиковий корпус. Робота за принципом "замкнений контакт – розімкнений контакт", сповіщувач дозволяє використовувати його в широкому колі за-вдань: контроль відкриття дверей, лічильники спрацьовувань/швидкості/частоти і т.д. рис.3.11.



Рисунок 3.11 – Магнітний сповіщувач

Бібліотеки . Бібліотека в Arduino – це програмний код в зовнішніх файлах, які мож-на встановити і підключити до вашого скетчу. У бібліотеці зберігаються різні методи і структури даних, які потрібні для спрощення роботи з датчиками, індикаторами, модулями та іншими компонентами. Використання бібліотек істотно спрощує роботу над проектами, тому що можна зосередитися на ос-новною логікою програми, не витрачаючи час на безліч дрібниць. Сьогодні величезна

кількість бібліотек викладено в інтернеті, де їх можна легко завантажити, причому абсолютно безкоштовно. З точки зору файлової системи бібліотека являє собою каталог, який містить певні папки. Під час компіляції і збірки проекту Arduino IDE автоматично включає в код ті класи, структури даних і методи з бібліотек, які включені і використовуються в скетчі. Таким чином, єдине, що нам потрібно зробити - це виставити в свій код відповідні інструкції, попередньо переконавшись, що потрібна бібліотека встановлена.

Щоб почати користуватися бібліотекою, потрібно на початку програми включити заголовний `h` файл директивою `include`. Наприклад, підключення бібліотеки `LiquidCrystal.h` буде виглядати так: `#include <LiquidCrystal.h>`

Кожна бібліотека повинна містити як мінімум 2 файли – це заголовний файл з розширенням `.h` і файл з вихідним кодом з розширенням `.cpp`. В заголовки знаходяться опис класу, константи і змінні. У другому файлі містяться коди методів. Крім двох основних файлів може міститися текстовий документ `keywords.txt` і папка `examples` з кодами прикладів використання бібліотеки. Файли `h` і `cpp` не обов'язково лежать в корені [25].

1). Встановити бібліотеку можна двома способами – за допомогою засобів Arduino IDE і вручну.

Щоб встановити бібліотеку за допомогою засобів Arduino IDE потрібно зайти в меню Скетч – Підключити бібліотеку – Додати .Zip бібліотеку рис.3.12.

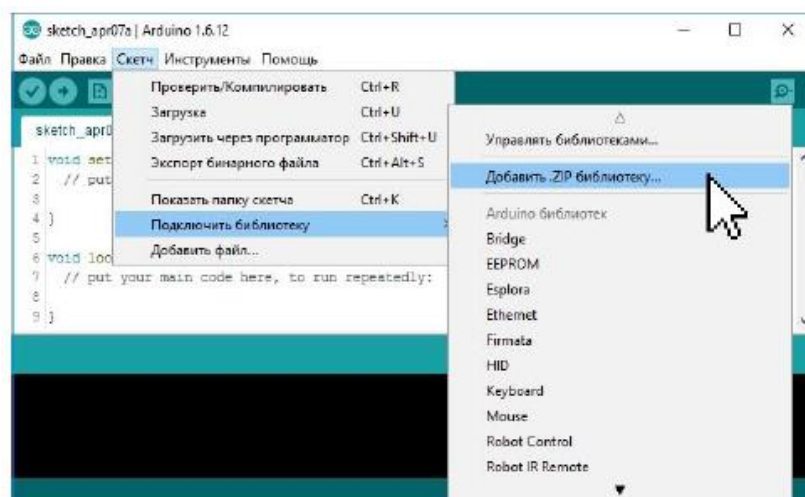


Рисунок 3.12 – Додавання бібліотеки

Для того щоб підключити бібліотеку, потрібно написати всього один рядок на початку скетчу: "#include<файл.h>" Потрібні бібліотеки:

- iarduino_GSM – бібліотека для роботи з GSM/GPRS Shield;
- OneWire – бібліотека для роботи з датчиками фірмиDallas;
- DallasTemperature – для роботи с цифровими термометрами DS18B20;
- SoftwareSerial.

Проектування системи UML-засобами. UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення.

UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю [26].

1). UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація. UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем. Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Ця діаграма є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання [27].

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою. Так, для об'єкту розробки діаграма прецедентів буде мати наступний вигляд рис.3.13.

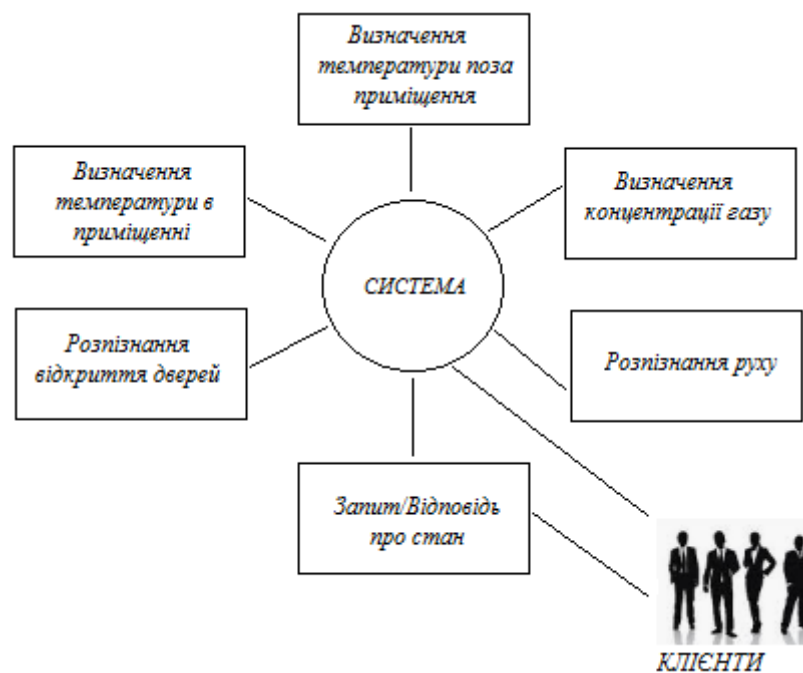


Рисунок 3.13 – Діаграма USE-CASE

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, включно із виконанням тих дій, що містяться в

діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності.

Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку [28]. Ці діаграми широко використовуються в описі поведінки, що включає велику кількість паралельних процесів.

Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції. Таким чином, діаграму діяльності можна вважати приватним випадком діаграми станів Основним напрямком використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно надати алгоритми їх виконання. Наступна діаграма демонструє основні етапи розробки системи безпеки будинку рис.3.14:

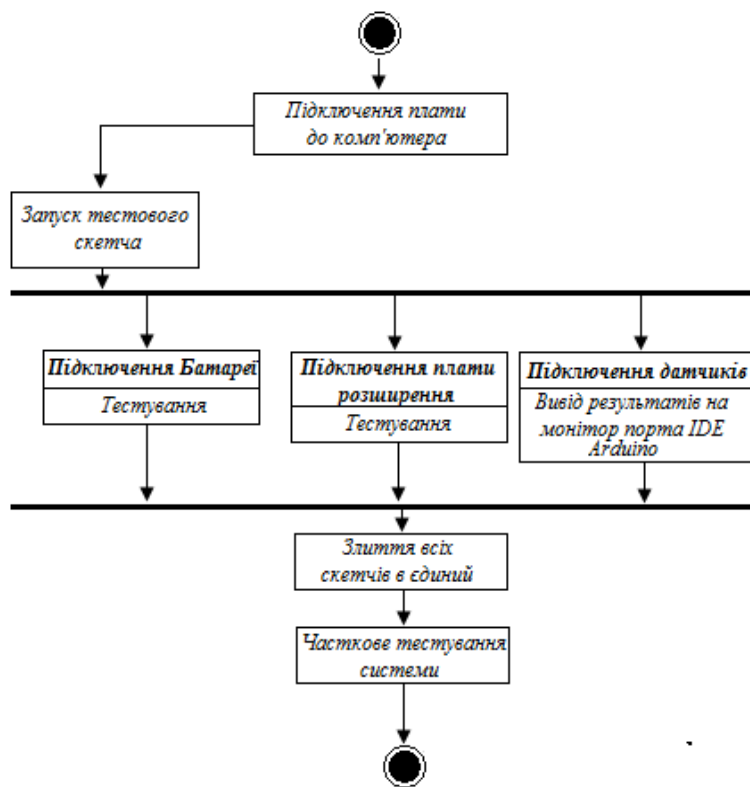


Рисунок 3.14 – Діаграма діяльності процесу розробки алгоритму роботи системи

3.3 Дослідження особливостей зборки елементів системи «Розумний будинок»

Основним мікроконтролером виступає Arduino Piranha Uno рис. 3.15.

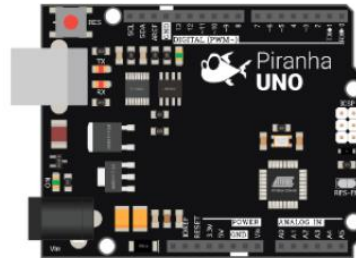


Рисунок 3.15 – Монтажна реалізація елементної бази плати Arduino Piranha Uno

Встановлюємо Battery Shield на Arduino Piranha UNO рис. 3.16:

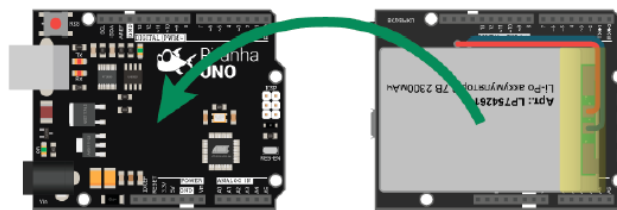


Рисунок 3.16 – Додавання елемента Battery Shield до Arduino Piranha UNO

Принцип підключення елемента живлення до макетної плати Arduino Piranha UNO аналогічен як і для елемента Battery Shield. Схема підключення розроблена за допомогою програми Fritzing і представлено на рис.3.17:

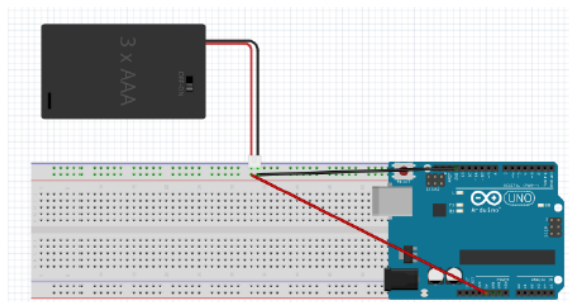


Рисунок 3.17 – Схема підключення батареї до плати Arduino Piranha UNO

Наступний елемент GSM/UMTS/LTE Shield A6 встановлюємо поверх Battery Shield рис.3.18.

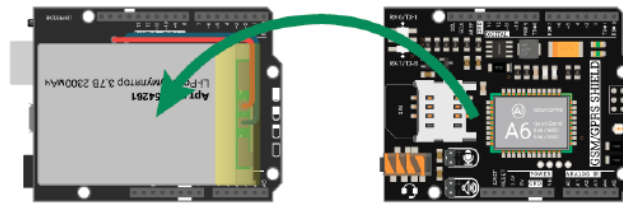


Рисунок 3.18 – Додавання елемента GSM/UMTS/LTE Shield A6 до плати Arduino Piranha UNO

На макетній платі використовується подібний елемент, який аналогічно підключається поверх плати Arduino Piranha UNO це зображено на рис. 3.19.

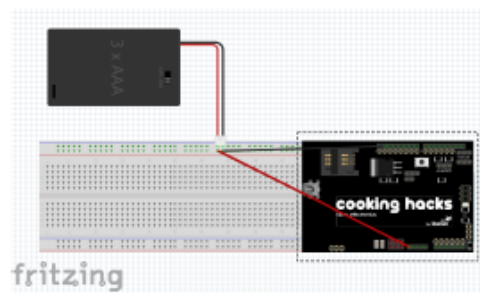


Рисунок 3.19 – Схема підключення GSM/UMTS/LTE Shield до плати Arduino Piranha UNO

Далі для зручності на GSM/UMTS/LTE Shield A6 встановлюємо Trema Shield рис.3.20.

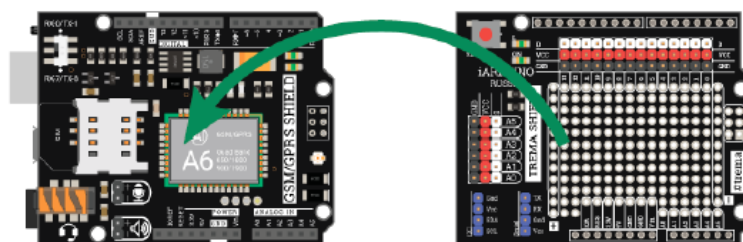


Рисунок 3.20 – Додавання плати розширення до Arduino Piranha UNO

Підключаємо гарнітуру до GSM/GPRS Shield A6. Також необхідно підключити цифровий термометр до Trema Shield рис.3.21.

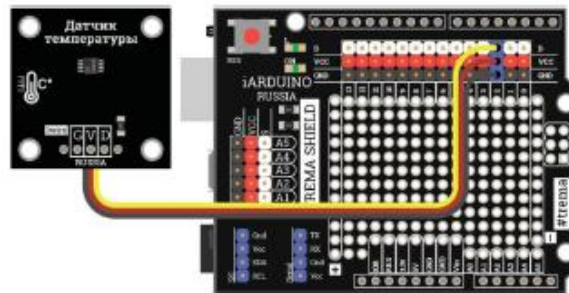


Рисунок 3.21 – Додавання цифрового термометру до Trema Shield

Наступним етапом буде підключення датчику газу до Trema Shield рис.3.22.

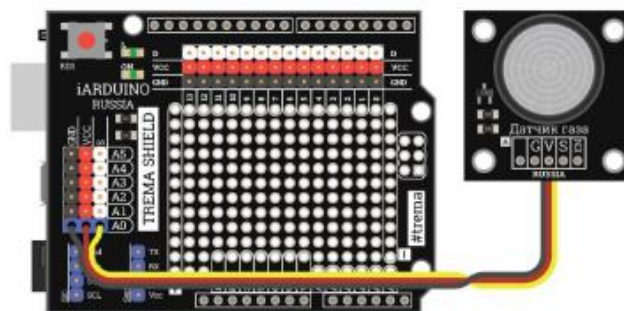


Рисунок 3.22 – Додавання датчику газу до Trema Shield

Наступним кроком підключаємо цифровий герметичний термометр до Trema Shield рис. 3.23. Міжінформаційним проводом (жовтий) і проводом живлення 5В (червоний) встановлено резистор 10кОм.

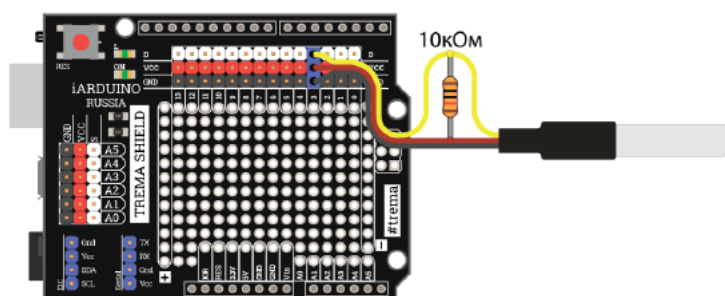


Рисунок 3.23 – Додавання зовнішнього термометру до Trema Shield

Підключаємо магнітний сповіщувач (геркон) до TremaShield рис.3.24. Від інформаційного дроту (жовтого) через резистор 10кОм підключається дріт GND (чорний).

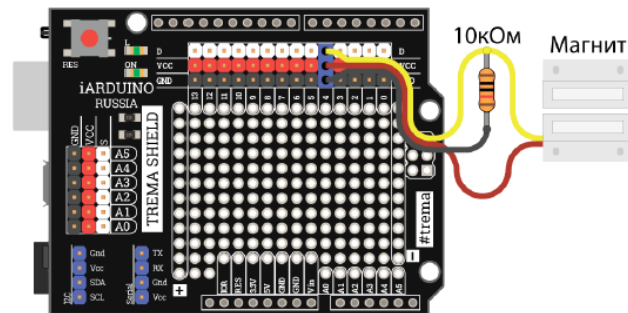


Рисунок 3.24 – Додавання магнітного сповіщувача до Trema Shield

3.4 Програмування пристроїв

Опис функціонування системи. Після увімкнення системи необхідно дочекатися СМС повідомлення про те, що ініціалізація пройшла успішно и модем GSM зареєструвався в мережі. Після цього система готова до роботи. Система може приймати повідомлення з командою "ОТЧЕТ". Після отримання такого повідомлення від користувача система сформує та відправить користувачу повідомлення з даними про стан системи та всіх датчиків. Також пристрій має кілька сценаріїв роботи з датчиками: при спрацюванні любого з датчиків на телефон користувача відправляється СМС з відповідним повідомленням, яке може містити також інформацію безпосередньо з конкретного датчика. При спрацюванні деяких особо важливих датчиків система безпеки відправляє СМС з відповідним повідомленням, а після цього здійснює дзвінок на вказаний користувачем номер, завдяки чому абонент може почути, що відбувається на об'єкті, який знаходиться під охороною.

Опис алгоритму роботи. З початку ми визначаємо та об'являємо змінні та функції, підключаємо бібліотеки, також створюємо об'єкти для подальшої роботи з ними. Одразу підключаємо бібліотеку для роботи з GSM/GPRS Shield та бібліотеку для програмної реалізації шини UART.


```
#include <iarduino_GSM.h>
```

```
#include <SoftwareSerial.h>
```

Наступні бібліотеки для роботи з OneWire та цифровим термометром DS18B20.

```
#include<OneWire.h>
```

```
#include<DallasTemperature.h>
```

Також створюємо об'єкти для подальшої роботи з датчиками температури.

```
OneWireoneWire_in(2);
```

```
OneWireoneWire_out(3);
```

```
DallasTemperaturesensor_inhouse(&oneWire_in);
```

```
DallasTemperaturesensor_outhouse(&oneWire_out);
```

Далі створюємо об'єкт GSM для роботи з функціями і методами бібліотеки iarduino_GSM, а також об'єкт softSerial для роботи по програмній шині UATR

```
iarduino_GSMgsm;
```

```
SoftwareSerialsoftSerial(7, 8);
```

Наступним кроком створюємо змінні Textдля зберігання текстової інформації та змінну пошуку символу у рядку.

```
StringText;
```

```
StringstrSMStxt;
```

Далі оголошуємо строку для зберігання тексту прийнятих повідомлень, строку для зберігання номеру відправника SMS повідомлень, а також у змінній number задаємо номер, з яким буде працювати система.

```
charSMStxt[161];
```

```
charSMSnum[13];
```

```
charnumber[13] = "+380978657912";
```

Оголошуємо масив для зберігання покажчика на рядок та Масив з номерами виводів пристроїв

```
char* Func = "ОТЧЕТ";
```

```
charPins[] = {A0, 4, 5};
```

Далі вказуємо мінімальну температуру спрацьовування для термометра всередині приміщення:

```
int8_t MIN_TEMP_IN = 15;
```

Максимальну температуру всередині

```
int8_t MAX_TEMP_IN = 28;
```

Мінімальну та максимальну температуру назовні

```
int8_t MIN_TEMP_OUT = 10;
```

```
int8_t MAX_TEMP_OUT = 35;
```

Максимальне значення концентрації газу:

```
int16_t MAX_GAS_VALUE = 450;
```

Наступні змінні відповідають за час, після закінчення якого проводиться повторна відправка СМС, час між відправленням СМС та тривожним дзвінком та тривалість дзвінка відповідно.

```
uint32_t delta_time = 100000;
```

```
uint32_t waiting_time = 5000;
```

```
uint32_t dial_waiting = 30000;
```

Далі вказуємо таймери для тривожного дзвінка, датчики температур, газу, руху, стану на повторне відправлення СМС. За замовчуванням всі вони дорівнюють 0.

```
uint32_t dial_time = 0;
```

```
uint32_t temp1_time = 0;
```

```
uint32_t temp2_time = 0;
```

```
uint32_t gas_time = 0;
```

```
uint32_t motion_time = 0;
```

```
uint32_t gerkon_time = 0;
```

```
uint32_t call = 0;
```

Функція `voidsetup()` містить конфігурацію обраних виводів, ініціалізацію роботи GSM/GPRS Shield. Також вона містить цикл очікування готовності GSM/GPRS Shield до роботи (реєстрації в мережі оператора). Після виконання всіх

указаних дій, виконується сповіщення о готовності до роботи, через відправку смс на вказаний номер.

```
void setup() {
  sensor_inhouse.begin();
  sensor_outhouse.begin();
  gsm.begin(softSerial);
```

Чекаємо завершення реєстрації модему в мережі оператора зв'язку.

```
while (gsm.status() != GSM_OK) { delay(1000); }
pinMode(Pins[0], INPUT);
pinMode(Pins[1], INPUT);
```

Установка кодування для символів кирилиці:

```
gsm.TXTsendCodingDetect("n");
```

Та відправляємо повідомлення про вдалий запуск:

```
gsm.SMSsend( F("Система активирована успешно."), number); }
```

Код `void loop()` виконується в циклі. При отриманні СМС або при виникненні події, описаної в коді, пристрій відправляє повідомлення абоненту або виконує тривожний дзвінок. Для того щоб після виникнення події модуль не почав постійно надсилати повідомлення чи робити дзвінки, була додана часова затримка `DELTA_TIME`, впродовж якої модуль ігнорує повторні події. Після закінчення цього часу, якщо нічого не змінилося, система повторно відправить СМС або здійснить тривожний дзвінок.

```
void loop () {
  Text = "";
```

Код програми побудований таким чином, що при видаленні одного з встановлених модулів система залишається повністю працездатною. Основну частину програми можна розділити на такі блоки: Датчик газу: у змінній `MAX_GAS_VOLUME` в самому початку ми задаємо гранично допустимий рівень газу в приміщенні. При досягненні цього значення модуль надійшло на вказаний номер СМС з повідомлення про те, що концентрація газу перевищила норму. Якщо

концентрація газу у нормі, то ці данні будуть внесені до строки Text, яка при запиті користувача буде вислана йому у СМС разом с показниками інших модулей.

```

if (analogRead(Pins[0]) >= MAX_GAS_VALUE) {
  if ((gas_time + delta_time) < millis() || gas_time == 0) {
    gas_time = millis();
    Text = "Превышена концентрация газа. \r\nЗначение=" + analogRead(Pins[0]);
  }
}

```

Далі відправляємо повідомлення:

```

gsm.SMSsend(Text, number); }
} Text = Text + "Газ: " + analogRead(Pins[0]) + "ppm";
delay(10);

```

Датчик температури внутрішній: вказавши в змінних MIN_TEMP_IN і MAX_TEMP_IN границі допустимих температур, налаштуємо модуль на ро-боту з цифровим термометром, розташованим всередині приміщення. При досягненні порогового значення модуль відправить на вказаний в змінної number номер СМС з повідомленням про те, що температура зміни-лася більше, ніж вказано в умові. Якщо температура всередині приміщення в нормі, то дані будуть внесені в рядок Text, який при запиті користувача буде висланий йому по СМС разом з показниками інших модулів.

```

sensor_inhouse.requestTemperatures();

if (sensor_inhouse.getTempCByIndex(0) <= MIN_TEMP_IN ||
    sensor_inhouse.getTempCByIndex(0) >= MAX_TEMP_IN) {
  if ((temp1_time + delta_time) < millis() || temp1_time == 0) {
    temp1_time = millis();
    Text = "Пороговое изменение температуры внутри. \r\nЗначение
    = " + String (sensor_inhouse.getTempCByIndex(0));
    gsm.SMSsend(Text, number); }
  } Text=Text+"\r\nТемпум:" + sensor_inhouse.getTempCByIndex(0);
  delay(10);
}

```

Датчик температури зовнішній: вказавши в змінних MIN_TEMP_OUT і MAX_TEMP_OUT границі припустимих температур, налаштуємо модуль на роботу з цифровим термометром, розташованим зовні приміщення. При досягненні порогового значення модуль відправить на вказаний в змінній number номер смс з повідомленням про те, що температура змінилася більше, ніж вказано в умові. Якщо температура зовні приміщення в нормі, то дані будуть внесені в рядок Text, який при запиті користувача буде висланий йому по СМС разом з показниками інших модулів.

```

sensor_outhouse.requestTemperatures();
if (sensor_outhouse.getTempCByIndex(0) <= MIN_TEMP_OUT //
    sensor_outhouse.getTempCByIndex(0) >= MAX_TEMP_OUT) {
if ((temp2_time + delta_time) < millis() // temp2_time == 0) {
temp2_time = millis();
Text = "Пороговое изменение температуры снару-жи.\r\nЗначение="+String
(sensor_outhouse.getTempCByIndex(0));; gsm.SMSsend(Text, number); }
} Text=Text+"\r\nТвнеш:"+sensor_outhouse.getTempCByIndex(0); delay(10);

```

Датчик руху: перевіряємо, чи немає руху в радіусі дії датчика. Якщо рух помічено, то модуль відправляє СМС абоненту на номер, вказаний у змінній number, далі чекає (waiting_time) і після цього робить тривожний дзвінок на той же номер. В змінній gsm.SOUNDdevice вказано параметр GSM_HEADSET, що означає, що на стороні модуля встановлена гарнітура.

Тривожний дзвінок можна налаштувати:

- модуль здійснює тривожний дзвінок "до переможного", поки або-нент не підніме трубку або не спрацює автовідповідач;
- модуль здійснює тривожний дзвінок, а після того, як трубка буде знята, обмежить розмову на час, вказаний в -змінній diall_waiting.

Якщо рух не помічено, то дані будуть внесені до рядка Text, який при запиті користувача буде вислано йому по СМС разом з показниками інших модулів.

```

if (digitalRead(Pins[1])) {
if ((motion_time + delta_time) < millis() // motion_time == 0) {

```

```

motion_time = millis();
gsm.SMSsend("Замечено движение!", number);
if ((call + waiting_time) < millis() || call == 0 ) {

```

Чекаємо, поки не пройде час `waiting_time`, перш ніж зробимо тривожний дзвінок.

```

gsm.SOUNDdevice(GSM_HEADSET);
if (gsm.CALLdial(number)) { while (gsm.CALLstatus() != GSM_CALL_ACTIVE)
{}

```

Чекаємо, поки статус дзвінка не зміниться на `GSM_CALL_ACTIVE`.

```

dial_time = millis();
while ((dial_time + dial_waiting) > millis() ) {}

```

Чекаємо час `dial_waiting`, перш ніж автоматично розірвемо з'єднання та розриваємо його.

```

gsm.CALLend();
} //gsm.CALLdial(number);
} } }
Text = Text + "\r\nДвижение: -";
delay(10);

```

Датчик стану (геркон): перевіряємо, чи немає розмикання датчика. Як-що датчик розімкнути, то модуль відправляє СМС абоненту на номер, вказаний у змінній `number`, далі чекає деякий час (`waiting_time`) і після цього робить тривожний дзвінок на той же номер. В змінної `gsm.SOUNDdevice` вказано параметр `GSM_HEADSET`, що означає, що на стороні модуля встановлена гарнітура. Тривожний дзвінок можна налаштувати: модуль здійснює тривожний дзвінок "до переможного", поки абонент не підніме трубку або не спрацює автовідповідач. Модуль здійснює тривожний дзвінок, а після того, як трубка буде знята, обмежить розмову на час, вказаний в змінній `dial_waiting`. Якщо датчик замкнутий, то дані будуть внесені в рядок `Text`, який при запиті користувача буде вислано йому по СМС разом з показниками інших модулів.

```

if (!digitalRead(Pins[2])) {

```

```

if ((gerkon_time + delta_time) < millis() || gerkon_time == 0) {
  gerkon_time = millis();
  gsm.SMSsend("Окно открыто!", number);
  if ((call + waiting_time) < millis() || call == 0) {

```

Чекаємо, поки не пройде час `waiting_time`, перш ніж зробимо тривожний дзвінок та робимо його.

```

gsm.SOUNDdevice(GSM_HEADSET);
if (gsm.CALLdial(number)) {
  while (gsm.CALLstatus() != GSM_CALL_ACTIVE) {}
  dial_time = millis();
  while ((dial_time + dial_waiting) > millis() ) {}

```

Чекаємо час `dial_waiting`, перш ніж автоматично розірвемо з'єднання та розриваємо його.

```

gsm.CALLend();
} //gsm.CALLdial(number);
} } }
Text = Text + "\r\nОкно: +";
delay(10);

```

Запит звіту: модуль перевіряє раз в секунду, чи немає вхідного повідомлення. Якщо повідомлення є, то модуль перевіряє, чи збігається номер, з якого прийшло повідомлення, з номером, вказаним в змінній `number`: Якщо номер збігається, то модуль перевіряє текст повідомлення: якщо в тексті повідомлення вказана команда з змінної `Func`, то модуль відправляє у відповідь СМС з повним звітом по всіх датчиках. Якщо в тексті СМС помилка, то модуль у відповідному СМС повідомить про це. У разі, якщо номер не збігається, то модуль буде ігнорувати текст СМС:

```

if (millis() % 1000 < 100) {
  delay(100);
  if (gsm.SMSavailable()) {
    gsm.SMSread(SMStxt, SMSnum);

```

```

if (!strcmp(number, SMSnum)) {
strSMStxt = SMStxt;
if (strSMStxt.indexOf(Func) != -1 ) {
gsm.SMSsend(Text, number);
} else {
gsm.SMSsend(F("Ошибка в тексте!"), number);
} } } }

```

Відповідь на вхідний дзвінок: модуль так само підтримує голосові вхідні дзвінки для того, щоб абонент зміг у будь-який час дізнатися, що відбувається у приміщенні, яке знаходиться під охороною. В змінної gsm.SOUNDdevice вказано параметр GSM_HEADSET, що означає, що на стороні модуля встановлена гарнітура. Для того, щоб модуль відповів на вхідний дзвінок, номер, з якого здійснюють виклик, повинен збігатися з номером, вказаним в змінної number.

```

if (gsm.CALLavailable(SMSnum)) {
if (!strcmp(number, SMSnum)) {
gsm.SOUNDdevice(GSM_HEADSET);
gsm.CALLup();
while (gsm.CALLavailable()) {}
} } }

```


ВИСНОВКИ

В результаті виконання бакалаврської кваліфікаційної роботи в першу чергу був проведений огляд принципів побудови та функціонування системи «Розумний дім». В ході даного огляду були розглянуті переваги і недоліки таких рішень як управління системою «Розумний будинок».

Наведено огляд досліджень, присвячених управлінню інтелектуальних будівлею, в тому числі стратегічного управління, а також управління підсистемою управління безпекою інтелектуальної будівлі. Наведено приклад автоматизованих системи управління.

У другому розділі бакалаврської роботи виконано формування теоретичних основ побудови систем управління розумним будинком. Для цього, перш за все, була сформована концепція побудови систем управління як соціотехнічної системи. На основі представленої концепції розроблена загальна технологія функціонування СУРБ.

Виконано аналіз формування вихідних даних, необхідних для функціонування СУРБ. Проведена системна класифікація факторів, що впливають на стан будинку, побудована загальна схема функціонування підсистеми збору та реєстрації даних в складі систем управління.

У практичній частині бакалаврської роботи виконана практична реалізація системи управління безпекою «Розумного будинку». Перш за все проведено дослідження алгоритму функціонування системи управління безпекою будинку.

Виконано вибір апаратних засобів та елементів системи. Після вибору апаратної бази були описані використовувані датчики і інше периферійне устаткування для моніторингу показників будинку. Далі було спроектовано розумну систему безпеки будинку на базі мікроконтролера Arduino, а також були розглянуті основні види систем безпеки, компоненти розумної системи безпеки, середовище проектування мікроконтролерів Arduino. У зв'язку з цим тематика бакалаврської роботи яка присвячена побудові інформаційної системи управління безпеки інтелектуальної будівлі, є актуальною.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гололобов В. Н. «Умный дом» своими руками / В. Н. Гололобов. - М. : НТ Пресс, 2017. - 416 с.
2. Е.А. Тесля. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Тесля Е.А. – Санкт Петербург, 2008. – 224 с.
3. Марк Э.С. Практические советы и решения по созданию "Умного дома", НТ Пресс, 2017 г.
4. Харке В. Умный дом. Объединение в сеть бытовой техники и системы коммуникаций в жилищном строительстве. Техносфера, 2016 г.
5. Гершкович В.Ф. Энергосберегающие системы жилых зданий: пособие по проектированию // С.О.К., 2018. № 8.
6. Нимич Г.В. Общие положения автоматического управления системами кондиционирования и вентиляции / Г.В. Нимич // С.О.К. – 2015. – № 7.
7. Спицын В.С., Спицын В.В. Серия . Компьютерные технологии, управление, радиоэлектроника. выпуск 17 // Алгоритмы управления температурой в помещениях, Вестник ЮУрГУ, 2012г. № 35. С. 79-84.
8. Harper R. Inside the Smart Home, ISBN-13: 978-1852336882, 2013. – 263с.
9. Гаскаров, Д.В. Интеллектуальные информационные системы/ Д.В. Гаскаров. –М.:Высшая школа, 2013. –432с
10. Богданов, С.В. Умный дом/ С.В. Богданов. –СПб.: Наука и техника 2015. –208 с.
11. Концепция системы «Умный Дом» —[Электронный Ресурс] — Режим доступа. — URL: <http://www.ascentis.ru/smart/smtheory/39-smtheorycon>
12. Carlini. Measuring a building IQ / J.A. Bernaden, et al (Ed.) // The Intelligent Building Sourcebook, Prentice-Hall, London, 2011, pp. 427 – 438.
13. W.F.E. Preiser, U. Schramm. Intelligent office building performance evaluation, Facilities 20 (7/8), 2012, pp. 279 – 287.

14. F.M. Lima, Intelligent building and its influence on the design process, Paper presented to the International Conference Sao Paulo, Oct. 25 – 26, 2013, Brazil: High Technology Buildings, Council on Tall Buildings and Urban Habitat, Brazil, 2013, pp. 139 – 149.

15. W.L. Chan, A.T.P. So, Intelligent Building Systems, Kluwer Academic Publishers, Boston, 2016.

16. W.M. Kroner. An intelligent and responsive architecture, Automation in Construction 6, 2016, pp. 381 – 393.

17. Ancevic. Intelligent building system for airport, ASHRAE Journal, 2015, pp.

18. Петин В. А. Создание умного дома на базе Arduino. – М.: ДМК Пресс, 2018.– 180 с.

19. Finch. Is IP everywhere the way ahead for building automation Facilities 19 (11/12) (2001), pp. 396 – 403.

20. L.M. Meade, A. Presley. R&D project selection using analytic network process, IEEE Transactions on Engineering Management 49 (1) 2002 (February), pp. 59 – 66.

21. M.I. Al Khalil. Selecting the appropriate project delivery method using AHP, International Journal of Project Management 20, 2002, pp. 469 – 474.

22. S. Smith. The integration of communications networks in the intelligent building, Automation in Construction 6, 1997, pp. 511 – 527.

23. Нгуен Суан Мань. Подсистема сбора и подготовки исходных данных в составе систем интеллектуального управления зданием / Нгуен Суан Мань Г.А. Попов, Е.И. Сироткина // Вестник АГТУ, No.3, 2015. с. 20-27.

24. Arduino ТОП-3 графических сред программирования CHIP. URL: <https://ichip.ru/arduino-top-3-graficheskikh-sred-programmirovaniya.html>

25. Среда разработки Arduino. URL: http://arduino.ru/Arduino_environment

26. Основы UML – диаграммы использования (use-case). URL: <https://pro-prof.com/archives/2594>.

27. UML UseCaseDiagramTutorial. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram>

28. Моделирование на UML. URL: <http://book.uml3.ua/>

ДОДАТКИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

БАКАЛАВРСЬКА РОБОТА

на тему:

**«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ
ПРИСТРОЯМИ РОЗУМНОГО БУДИНКУ»**

Виконав студент

Гайдайчук А.К.

Науковий керівник

Бондарчук А.П..

МЕТА ТА ЗАВДАННЯ БАКАЛАВРСЬКОЇ РОБОТИ

2

Мета роботи – побудова інформаційної системи управління безпеки інтелектуальної будівлі.

Об'єкт дослідження – підсистема управління «Розумного будинку»

Предмет дослідження – процес управління безпекою інтелектуального будинку.

Завдання бакалаврської роботи:

- 1. Аналіз сучасного стану систем управління інтелектуальною будівлею.*
- 2. Формування теоретичних основ побудови систем управління розумним будинком.*
- 3. Практична реалізація управління систем безпеки інтелектуального будинку*



Рис. 1. Основні компоненти «розумного будинку»



Рис. 2. Основні підсистеми «розумного дому»

Сьогодні складно уявити світ без автоматизації. Житловий будинок не виняток. Можна автоматизувати все, від вклучення світла до управління мікрокліматом в кімнаті. Дана можливість з'явилася завдяки системі «Розумний будинок». Такі системи бувають дуже різноманітними але виконують корисну роль, а саме: захищають від несанкціонованого доступу сторонніх осіб, контроль стану протипожежної безпеки, витік газу та інше.

СТРУКТУРА СИСТЕМИ «РОЗУМНИЙ БУДИНОК»



Рис. 1. Структурна схема системи «Умный дом»

1. Компоненти нижнього рівня



2. Компоненти середнього рівня



3. Компоненти верхнього рівня



Структура системи «Розумний будинок» складається з трьох рівнів, а саме нижнього, середнього і верхнього. Нижній рівень, це датчики і виконавчі елементи, такі як сервоприводи, датчики вологості і температури, ультразвукові сенсори тощо. Середній рівень системи «Розумний будинок» в нашому випадку представлений у вигляді контролера Arduino. Верхній рівень, це програмне забезпечення або ж Web-додаток через яке здійснюється управління системою «Розумний будинок».



Рис.1. Сенсорна панель



Рис.2. Пульти дистанційного управління



Рис.3. Смартфон



а - Ocelot (X10), б - TVLINK 868 GSMAL (Teleco), в - Neroll 8450-50 (Nero)

Рис.4. Багатофункціональні контролери в корпусному виконанні

Управляти розумним будинком можна і за допомогою стаціонарного телефону. Наприклад, використовуючи домофон, можна влаштувати так, що подзвонив у двері людина ні в якому разі не запідозрить, що вас немає вдома. В даному випадку Другий вид управління на базі промислових контролерів. Як приклад можна навести деякі багатофункціональні контролери, приклад яких зображено на рис.4. слайд 5.

КОНЦЕПЦІЯ ПОБУДОВИ СИСТЕМИ УПРАВЛІННЯ ІНТЕЛЕКТУАЛЬНОЮ БУДІВЛЕЮ

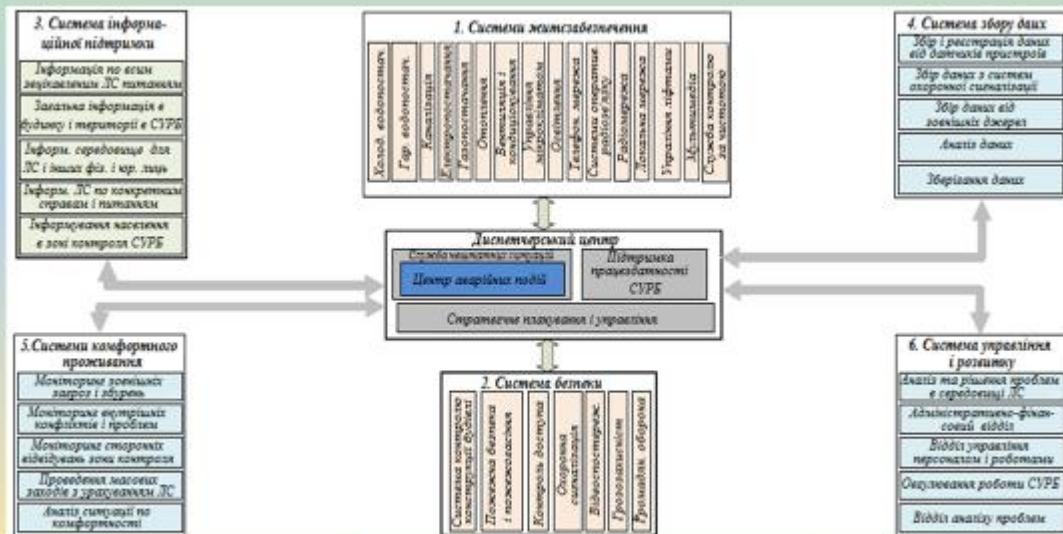


Рис.1. Концепція систем управління розумним будинком



Рис.1. Інтерфейс ПЗ NetPing



Рис.2. Інтерфейс ПЗ Home Sapiens



Рис.3. Інтерфейс ПЗ MajorDoMo



Рис.2. Інтерфейс ПЗ Fibaro

Табл.1. Порівняльні характеристики програмно-апаратних рішень

	NetPing	Open Remote	Home Sapiens	Major DoMo	Fibaro	Arduino
Простота налаштування	+	-	+	+	+	+
Відкритість системи	-	+	-	+	-	+
Мобільні додатки	+	+	+	+	+	+
WEB інтерфейс	+	+	+	+	+	+

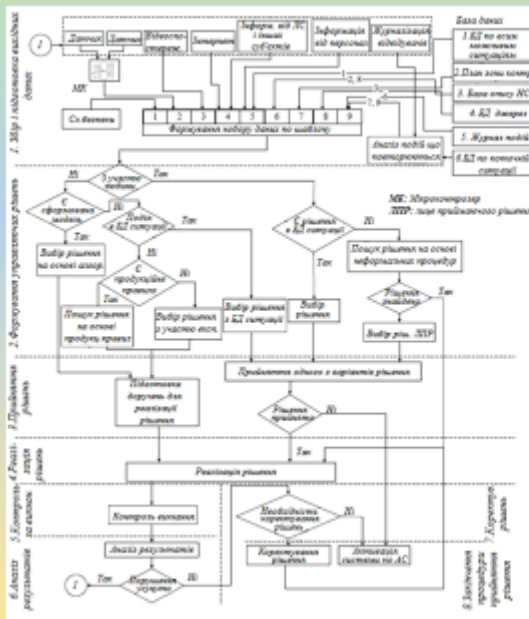


Рис.1. Загальний алгоритм процесу прийняття рішень в СУРБ

Алгоритм складається з наступних етапів:

Перший етап, збір і підготовка даних, націлена на формування образу поточної ситуації відповідно до встановленого шаблону ситуації.

Другий етап - формування управлінських рішень - є одним з основних. Залежно від ситуації передбачені різні варіанти вироблення керуючих рішень з урахуванням змісту даних, що надійшли.

Третій етап пов'язаний безпосередньо з актом прийняття певного рішення. Наступні етапи реалізації рішення і контролю типові для будь-яких систем організаційного управління, і тому їх деталізація не наводиться. На основі дослідженого загального алгоритму прийняття рішень був проведений аналіз складу інформаційних технологій, необхідних для реалізації цього алгоритму. Були виявлені 13 типів інформаційних технологій, які більш детально описані у другому розділі бакалаврської роботи.

Набір інформаційних технологій в складі типовий СУРБ :

1. Для ІТ 1: Hewlett-Packard, VEE 3.0, IAR EWb; 2. Для ІТ 2: програма Сайт Супутнику, Resuva; 3. Для ІТ 3: програма WV-ASM200, Secur OS; 4. Для ІТ 4: Zoho Docs, XaitPorter, Process Street 5. Для ІТ 5: Profive®; 6. Для ІТ 6: master's degree; 7. Для ІТ 7: SCADA 8. Для ІТ 8: Accident Alarm System (CAAS); 9. Для ІТ 9: SAM-Method, EMO-method. 10. Для ІТ 11: SCADA systems. Для типів 10, 12, 13 відповідних програмних продуктів, які могли б бути використані в СУРБ, знайти не вдалося.

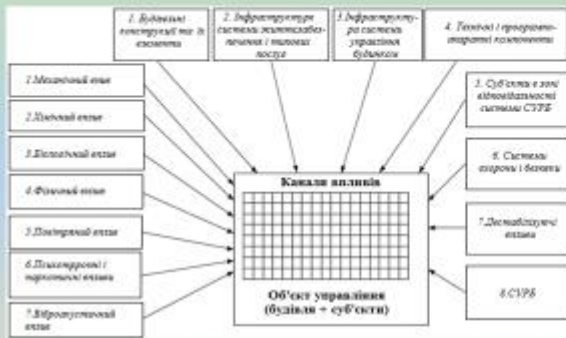


Рис.1. Діаграма каналів впливу на об'єкт управління



Рис.2. Алгоритм роботи системи контролю

Класи можливих способів впливу на систему управління :

1. Механічні дії.
2. Хімічний вплив.
3. Біологічний вплив.
4. Фізичний вплив.
5. Повітряний вплив.
6. Психотропні та наркотичні впливи.
7. Вібродокументальний вплив

Проаналізовано роботу алгоритму контролю безпеки в системі «Розумний будинок», який зображено на рис.2. Контроль здійснюється на основі моніторингу параметрів від показників датчиків. Датчики встановлюються в приміщенні. Інформація з датчиків надходить на локальні контролери, які її обробляють і на основі отриманих даних здійснюють регулюючий вплив. Як видно з функціональної схеми, користувачі безпосередньо можуть задавати параметри локальним контролерам і отримувати від них необхідну інформацію.

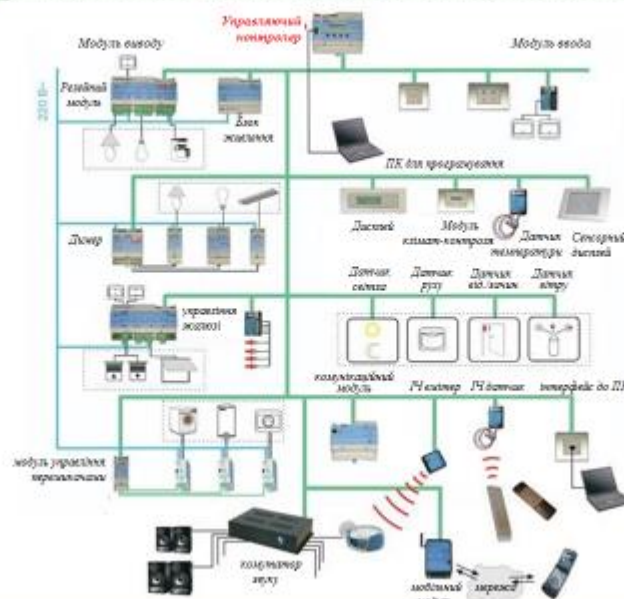




Рис.1.

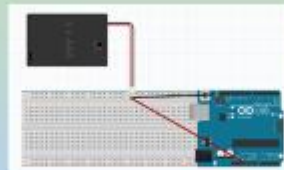


Рис.2.



Рис.3.



Рис.4.



Рис.5.

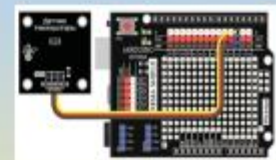


Рис.6.

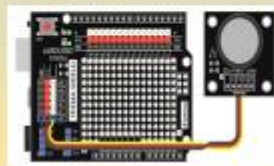


Рис.7.

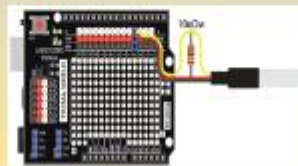


Рис.8.

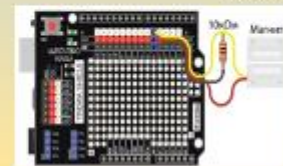


Рис.9.

ПРОГРАМУВАННЯ ПРИСТРОЇВ

12

З початку ми визначимо та об'явимо змінні та функції, підключаємо бібліотеки, також створюємо об'єкти для подальшої роботи з ними. Одразу підключаємо бібліотеку для роботи з GSM/GPRS Shield та бібліотеку для програмної реалізації шини UART.

```
#include <arduino_GSM.h>
#include <SoftwareSerial.h>
```

Створюємо об'єкти для подальшої роботи з датчиками температури

```
OneWireoneWire_in(2);
OneWireoneWire_out(3);
DallasTemperatureSensor_inhouse(&oneWire_in);
DallasTemperatureSensor_outhouse(&oneWire_out);
```

Далі створюємо об'єкт GSM для роботи з функціями і методами бібліотеки `arduino_GSM`, а також об'єкт `softSerial` для роботи по програмній шині UART.

```
arduino_GSMgsm;
SoftwareSerialsoftSerial(7, 8);
```

Наступним кроком створюємо змінні `Text` для зберігання текстової інформації та змінну пошуку символу у рядку.

```
StringText;
StringstrSMStxt;
```

Далі вказуємо мінімальну і максимальну температуру спрацювання для термометра всередині приміщення:

```
int8_t MIN_TEMP_IN = 15;
int8_t MAX_TEMP_IN = 28;
```

Мінімальну та максимальну температуру назовні

```
int8_t MIN_TEMP_OUT = 10;
int8_t MAX_TEMP_OUT = 35;
```

Отже, підсумовуючи вищесказане можливо зробити висновок:

- проведений аналіз показує актуальність та перспективи застосування системи «Розумний будинок».
- в ході аналізу принципів побудови та функціонування системи були показані переваги і недоліки таких рішень як управління системою «Розумний будинок».
- була сформована концепція побудови систем управління як соціотехнічної системи. На основі представленої концепції розроблена загальна технологія функціонування СУРБ. Виконано аналіз формування вихідних даних, необхідних для функціонування СУРБ. Проведена системна класифікація факторів, що впливають на стан будинку, побудована загальна схема функціонування підсистеми збору та реєстрації даних в складі систем управління.
- виконана практична реалізація системи управління безпекою «Розумного будинку»

Отримані результати свідчать про доцільність використання розглянутих систем у побуті, а тематика бакалаврської роботи актуальною і своєчасною.