DOI: https://doi.org/10.36910/6775-2524-0560-2023-53-43
УДК 681.3.05:004.056
Розломій Інна Олександрівна, к.т.н., старш. викладач
https://orcid.org/0000-0001-5065-9004
Косенюк Григорій Володимирович, к.т.н., доцент
https://orcid.org/0000-0003-2103-3904
Науменко Сергій Васильович, аспірант
https://orcid.org/0000-0002-6337-1605
Михайловський Павло Васильович, аспірант
https://orcid.org/0009-0008-4324-1724
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси, Україна

МОДЕЛЮВАННЯ СИСТЕМИ ДАТЧИКІВ НА БАЗІ МІКРОКОНТРОЛЕРА В ІГРОВІЙ СИМУЛЯЦІЇ «СМАРТ-БУДИНОК» З ВИКОРИСТАННЯМ ШИФРУВАННЯ ДАНИХ

Розломій І.О., Косенюк Г.В., Науменко С.В., Михайловський П.В. Моделювання системи датчиків на базі мікроконтролера в ігровій симуляції «смарт-будинок» з використанням шифрування даних. Стаття описує процес розробки моделі системи датчиків, що заснована на мікроконтролері, здатної симулювати роботу «смарт-будинку». Технологія «смарт-будинку» стає все більш актуальною, оскільки забезпечує зручність та безпеку життя в приватному будинку або офісі. Із застосуванням датчиків, мікроконтролерів та інших пристроїв можливо автоматизувати процеси, що забезпечують комфорт, енергозбереження та зниження ризику аварійних ситуацій.Особлива увага приділяється використанню вбудованого модуля шифрування, що дозволяє захистити дані, що надходять з датчиків від несанкціонованого доступу. В сучасному світі зростає популярність розумних будинків та їх інтеграція з інтернетом речей. Однак, збільшення кількості підключених пристроїв та передача великого обсягу приватних даних ставлять під загрозу безпеку цих систем. У цьому контексті використання шифрування даних стає ключовим елементом для забезпечення конфіденційності та цілісності інформації. Дана стаття присвячена моделюванню системи датчиків на базі мікроконтролера для ігрової симуляції «смарт-будинок» та акцентує увагу на використанні вбудованого модуля шифрування. Застосування цього модуля дозволяє ефективно захищати дані, що надходять з датчиків, від несанкціонованого доступу. У статті розглядаються різні аспекти моделювання системи датчиків, включаючи вибір підходів до інтеграції мікроконтролера, розробку алгоритмів збору та обробки даних, а також використання шифрування для захисту інформації. Захист даних, отриманих з датчиків, стає надзвичайно важливим аспектом, і саме тому використання шифрування є ключовим елементом для забезпечення конфіденційності та цілісності інформації. Результати дослідження підтверджують ефективність використання вбудованого модуля шифрування в контексті «смарт-будинку» та його здатність забезпечувати безпеку даних, що передаються в систему.
Ключові слова: смарт-будинок, мікроконтролер Arduino, захист інформації, шифрування, ідентифікація, ігрова симуляція, ігровий рушій Unity.

Rozlomii I., Kosenyuk H., Naumenko S., Mikhailovsky P. Modeling a microcontroller-based sensor system in the game simulation «Smart-Home» using data encryption. The article describes the process of developing a model of a sensor system based on a microcontroller capable of simulating the operation of a «Smart Home». «Smart home» technology is becoming more and more relevant, as it provides convenience and safety of life in a private home or office. With the use of sensors, microcontrollers and other devices, it is possible to automate processes that provide comfort, energy savings and reduce the risk of emergency situations. Special attention is paid to the use of the built-in encryption module, which allows you to protect the data coming from the sensors from unauthorized access. In today's world, the popularity of smart homes and their integration with the Internet of Things is growing. However, the increase in the number of connected devices and the transfer of a large amount of private data puts the security of these systems at risk. In this context, the use of data encryption becomes a key element to ensure the confidentiality and integrity of information. This article is devoted to the modeling of a microcontroller-based sensor system for the game simulation «Smart Home» and focuses on the use of the built-in encryption module. The use of this module allows you to effectively protect the data coming from the sensors from unauthorized access. The article discusses various aspects of sensor system modeling, including the choice of microcontroller integration approaches, the development of data acquisition and processing algorithms, and the use of encryption to protect information. The protection of sensor data is becoming an extremely important aspect, and that is why the use of encryption is a key element to ensure the confidentiality and integrity of the information. The results of the study confirm the effectiveness of using the built-in encryption module in the context of the «Smart Home» and its ability to ensure the security of data transmitted to the system.
Key words: smart home, Arduino microcontroller, information security, encryption, identification, game simulation, Unity game engine.

Постановка проблеми та її зв'язок із важливими науковими чи практичними завданнями. Популярність домашньої автоматизації значно зросла в останні роки через більш високу доступність і простоту. Маючи можливість контролювати аспекти наших будинків і мати можливість автоматично реагувати на події, система стає все більш популярною і необхідною через міркування безпеки і затрат. Таким чином, «смарт-будинки» призначені забезпечити безпеку

будинку, захистити від будь-яких надзвичайних ситуацій [1]. Система безпеки «розумного дому» має насамперед виконувати функції захисту від вторгнення сторонніх осіб, автоматизації дверей, воріт, охоронної сигналізації, запобігання аварійним ситуаціям. У разі будь-якого займання або задимлення спрацює пожежна сигналізація. Про протікання води чи підвищеному рівні вологості система відразу повідомить господаря та відповідні служби.

Актуальність технології «смарт-будинок» полягає в її можливості забезпечувати комфорт та безпеку мешканців, а також економію енергоресурсів. Завдяки системі керування, користувач може контролювати пристрої в будинку з будь-якого місця та в будь-який час, що дозволяє забезпечити безпеку та зручність. Крім того, використання технології «смарт-будинок» дозволяє значно зменшити споживання електроенергії та інших ресурсів, що є важливим аспектом екології та сталого розвитку. Система «смарт-будинок» може включати в себе різноманітні пристрої, такі як датчики відкриття вікон, дверей та руху, камери спостереження, системи опалення та кондиціонування повітря, освітлення та інше [2-3]. Всі ці пристрої можуть бути підключені до однієї системи керування, що дозволяє забезпечити їх взаємодію та координацію.

Аналіз останніх досліджень та публікацій. Системи «смарт-будинок» стають все більш популярними, що вимагає забезпечення їх надійності та безпеки. З розвитком технологій «смарт-будинків» та інтернету речей, стає все більш очевидним значення створення безпечних та захищених систем збору та передачі даних. У таких системах, датчики відіграють ключову роль, забезпечуючи збір різноманітної інформації про оточуюче середовище та стан пристроїв у будинку. Однак, із збільшенням обсягу даних, котрі надходять з датчиків, виникає необхідність в їхньому надійному захисті від несанкціонованого доступу [4-6].

Інтерес науковців до захисту даних з датчиків систем «смарт-будинку» обумовлений зростаючою популярністю та поширеністю цих систем в сучасному житті. Завдяки «смарт-будинкам» користувачі можуть зручно керувати різними аспектами своїх домівок, такими як освітлення, опалення, безпека та енергоефективність, за допомогою мобільних пристроїв чи голосових асистентів [7].

Однак, збільшення кількості підключених пристроїв та обмін даними між ними створює нові виклики щодо захисту приватності та безпеки. Науковці [8, 11] виявляють інтерес до захисту цих даних, розробляючи криптографічні рішення та алгоритми шифрування, які забезпечують конфіденційність, цілісність та аутентичність інформації. Вони досліджують методи аутентифікації та авторизації, застосовуючи різні протоколи та механізми для забезпечення безпеки комунікацій та обміну даними в «смарт-будинках».

Вирішення проблеми довіри між мережею кожен та блок керування шляхом використання короткого маркера автентифікації із встановленням безпечного ключа сеансу представлено в роботі [9]. Запропонована схема дозволяє уникнути різноманітних популярних атакам, таким як атаки на відмову в обслуговуванні та атаки підслуховування.

Авторами [10] висвітлено різні вразливості безпеки розумних будинків на основі IoT, представити ризики для мешканців будинку та запропонувати підходи до пом'якшення виявлених ризиків. Результати дослідження можуть бути використані як основа для покращення вимог до безпеки розумних будинків на основі IoT.

Метою статті є дослідження особливостей побудови ефективної моделі системи датчиків для ігрової симуляції «смарт-будинок», що базується на мікроконтролері та використання вбудованого модуля шифрування для захисту надійності даних, отриманих з різних датчиків. Розглядається важливий аспект безпеки даних – використання вбудованого модуля шифрування. Його застосування дозволяє надійно захистити дані, що надходять з датчиків, та забезпечити конфіденційність і цілісність інформації, що передається у систему.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Система розумного дому передбачає автоматичне керування електронними пристроями в будинку, які підключені до Інтернету, що дозволяє дистанційно управляти ними. Крім стандартних функцій, таких як увімкнення опалення та охолодження, системи «смарт-будинків» також підвищують безпеку за допомогою камер, датчиків руху та датчиків вогню. Ці системи працюють через мережу пристроїв, які використовують різні протоколи зв'язку, такі як Wi-Fi, Bluetooth, ZigBee. Керування пристроями може здійснюватися дистанційно за допомогою контролерів, голосових помічників або додатків. Багато пристроїв мають вбудовані датчики, які відстежують зміни в русі, температурі та освітленні, щоб користувач мав змогу отримувати інформацію про

навколишнє середовище.

Домашня автоматизація має значну перевагу в тому, що вона надає власникам будинків спокій, дозволяючи їм дистанційно контролювати своє житло і запобігати ризикам, таким як забуті кавоварки, незамкнені двері, відкриті вікна або залишена увімкненою праска [11]. Однак, наявні й певні недоліки використання систем «смарт-будинків», зокрема, проблеми з безпекою та захищеністю даних користувачів, але з роками протоколи безпеки постійно вдосконалюються, зробивши ці технології все більш безпечними.

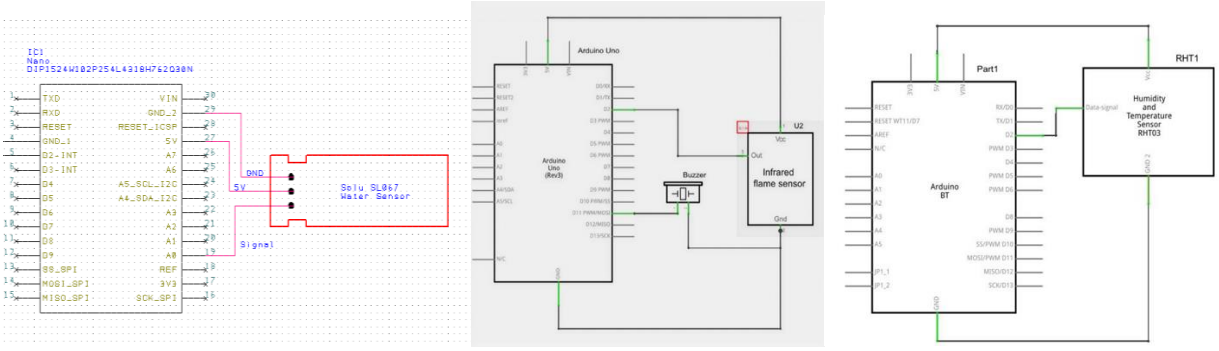
Сучасні інструменти для симуляції, зокрема, графічний двигун Unity, дозволяють моделювати складні системи, такі як «смарт-будинки», з різними датчиками, реле, мікроконтролерами та іншими пристроями. У симуляції «смарт-будинку» на базі мікроконтролера можуть бути підключені датчики вогню, води, дверей та вікон. Цей підхід дозволяє моделювати різні сценарії, такі як пожежа, протікання води, відкриті двері або вікна, що відображають поведінку системи у реальному часі.

Система датчиків смарт-будинку повинна мати модуль реєстрації та авторизації користувача для збереження даних і налаштувань. Вона повинна бути підв'язана до сервісу збереження базових даних, таких як електронна пошта та пароль, а також модель даних налаштувань. Користувачам має бути надана можливість відновлення паролю через електронну пошту. Система повинна мати зручний інтуїтивно зрозумілий інтерфейс для налаштувань, включаючи окреме налаштування датчиків, інформувань і особистого профілю. Користувачі повинні отримувати Push-повідомлення та повідомлення на електронну пошту. Датчики моніторингу мають інформувати користувача миттєво, якщо відхиляються встановлені норми. Система повинна містити датчики для контролю рівня води, температури, вологості та вогню. Для демонстрації роботи системи розробляється симуляція будинку, що відтворює надзвичайні ситуації, які викликають відповідні служби для усунення проблем.

Для функціонування системи потрібні наступні пристрої: пристрій на базі мікроконтролеру для підключення датчиків та зв'язку з ігровою симуляцією, датчик вогню, датчик температури та вологості, датчик рівня води і датчик для виявлення проникнення у будинок. Для демонстрації роботи системи використовується простий елемент – кнопка, підключену до мікроконтролера.

При спрацьовуванні датчика в системі, дані передаються на контролер. Контролер обробляє цю інформацію за допомогою програмного коду, вбудованого у прошивку. Після обробки дані надсилаються до додатку, який відтворює симуляцію смарт-будинку. Симуляція враховує користувацькі налаштування та візуалізує надзвичайні ситуації, такі як пожежа, затоплення або підвищений рівень вологості.

Пристрій складається з датчиків, підключених до мікроконтролера. У якості управляючого елементу пристрою використано плату Arduino Uno R3, до котрої підключені датчики води, вогню, температури та вологості, рис. 1.



а) б) в)

Рис. 1. Схеми підключення датчиків а) – води, б) – вогню, в) – температури та вологості до плати Arduino

Програмна частина додатку включає процес налаштування датчиків користувачем. Для збереження цих налаштувань використовується база даних, яка підключена до додатку. База даних має можливість зберігати дані авторизації та визначену структуру для налаштувань датчиків. Крім того, встановлено зв'язок між базою даних та середовищем Unity.

В якості серверної платформи використовується PlayFab. PlayFab є повноцінною серверною платформою для онлайн-ігор, що надає керовані ігрові сервіси та аналітику в реальному часі.

В якості шаблону проектування для створення та управління User Interface використовується Model-View-ViewModel. MVVM, як і інші шаблони проектування, сприяє організації коду і поділу програм на модулі, що спрощує розробку, оновлення та повторне використання коду, забезпечуючи більшу простоту та ефективність.

Для обробки даних з різних датчиків використовується паттерн Publisher-Subscriber.У шаблоні обміну повідомленнями Publisher-Subscriber видавці не надсилають повідомлення безпосередньо всім підписникам. Замість цього повідомлення передаються через «брокерів». Видавці не знають, хто є підписниками або на які події вони підписані, якщо такі є. Це означає, що операції видавця та підписника можуть працювати незалежно одна від одної. Цей підхід також називають слабким зв'язком, оскільки він усуває залежності між об'єктами, які інакше могли бути присутніми в традиційних шаблонах обміну повідомленнями.

Шаблон Publisher-Subscriber відрізняється від стандартних моделей запитів/відповідей тим, що видавці не перевіряють, чи доступні нові дані. Це є основою для ефективної потокової передачі даних у реальному часі. Використання шаблону Publisher-Subscriber дозволяє створювати динамічні мережі в масштабі, уникнувши перевантаження видавничих компонентів та непотрібних витрат.

Типові сценарії використання цього шаблону включають обмін повідомленнями про події, миттєві повідомлення та потокове передавання даних. Publisher-Subscriber також знаходить застосування в балансуванні робочого навантаження та асинхронних робочих процесах, рис. 2.

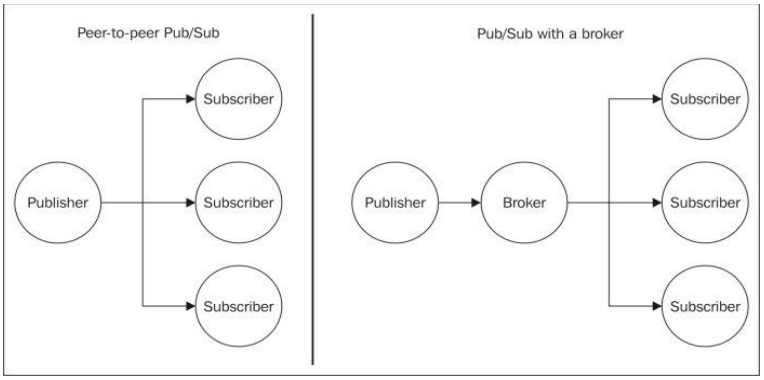


Рис. 2. Модель шаблону програмування Publisher – Subscriber

Для реалізації серверної частини потрібно створити обліковий запис PlayFab. Платформа надає широкий спектр сервісів, таких як гравці, економіка, таблиці лідерів та керування контентом. Ці послуги охоплюють облікові записи, торгівлю/валюти, таблиці лідерів та інші функції, наприклад, управління інвентарем. Для інтеграції PlayFab в проект Unity також потрібно встановити два компоненти: пакет PlayFab Unity SDK і пакет розширень PlayFab Editor. Це забезпечить необхідну функціональність для роботи з платформою у середовищі Unity. Для перевірки правильності налаштувань PlayFab створено нового гравця, дані якого автоматично заносяться до бази даних створеного проекту. Роль API виконує створений окремий клас з запитами до сервісу PlayFab. Методи авторизації приймають вхідні дані, такі як електронна пошта, пароль і нікнейм користувача, і створюють спеціальний запит у вигляді моделі. Для реєстрації користувачів використовується модель RegisterPlayFabUserRequest, а для входу під уже створеним обліковим записом – модель LoginWithEmailAddressRequest. Виклик API вимагає передачі двох зворотних викликів – для успішного виконання запиту і для обробки помилок.

Реалізація програмної частини у середовищі Unity розпочинається з імплементації базового класу DependentMonoBehaviour, який забезпечує реалізацію Dependency Injection паттерна. Цей базовий клас має велике значення в проекті, оскільки на етапі створення базових модулів програми потрібно зв'язувати ці модулі, дотримуючись принципів SOLID програмування.

З описаного базового класу походять два класи: UIController і UIPresentationModel<T>. Вони використовуються для реалізації шаблону проектування MVVM. З цих класів створюються необхідні класи для управління датчиками вогню, води, дверей, температури і вологості. Крім того, існують класи, які успадковуються від цих класів і відповідають за процес авторизації користувачів,

такі як RegisterForm і LoginForm.

Кожен з контролерів для датчиків реалізує інтерфейс ISensorSignalReceiver, який містить метод з типом датчика. Цей метод викликається у SensorsSignalReceiver, що втілює шаблон програмування Publisher-Subscriber.

Для роботи процесу авторизації створений клас LoginAPI, який має методи для обробки запитів з сервісом PlayFab. Цей клас використовується з LoginForm і RegisterForm, які отримують дані користувача з полів, що обробляються у View частині.

Розробка та складання апаратної частини здійснена на базі плати Arduino. Для збирання пристрою підготовлено наступні компоненти: макетна плата, датчик рівня води з вбудованим резистором, датчик температури та вологості з вбудованим резистором, датчик вогню, кнопка, резистори для кнопки та датчика вогню, перемички, плата Arduino.

Спочатку виконується підключення макетної плати. Вихід 3,3V або 5V з'єднується перемичкою до входу «плюс», залежно від вимог. Аналогічно з'єднуються виходи для заземлення (GND), що забезпечує живлення всієї плати.

Наступним кроком є підключення датчиків води, температури та вологості. Обидва датчика мають три виходи: заземлення, живлення та сигнал. Необхідні контакти з'єднуються з лінією «плюс» та «мінус» на макетній платі, а сигнал підводиться на аналогові входи плати Arduino. Оскільки датчики мають вбудовані резистори, додаткове підключення не потрібне.

Для кнопки і датчика вогню використовуються два резистори. Датчик вогню встановлюється на плату. Одна частина резистора та перемичка для аналогового сигналу підключаються до «плюса» датчика. Живлення підключається до іншої частини резистора. Заземлення підводиться до коротшої ніжки датчика. Щодо кнопки, один кінець резистора підключається до нижньої лівої ніжки кнопки.

Unity використаний як кросплатформний ігровий двигун, що дозволяє розробникам створювати ігри для Android та iOS з використанням однієї кодової бази. Для сценаріїв використана мова C#, яка не підтримується на Android та iOS. Однак код C# перетворений в код C++ за допомогою компілятора Unity Il2Cpp, який використовується для створення проектів Xcode та Android Studio.

Поміж коду, C++ Il2Cpp також генерує метадані, які зберігаютьсяу зовнішньому файлі під назвою «global-metadata.dat». Цей файл включений до згенерованої програми і завантажується середовищем виконання Il2Cpp під час ініціалізації програми. Він містить інформацію про всі об'єкти середовища виконання, включаючи імена, типи класів, методи, властивості та зв'язки між ними, а також рядкові літерали, такі як ключі API, які використовуються в коді C#. Ці метадані посилалися зі згенерованого коду C++ та використовуються для зв'язування методів з їх реалізацією.

Усі метадані, які були збережені у файлі «global-metadata.dat», легко отримати, що надає зловмисникам інформацію, яку вони можуть використати для зламування ігор.

Оскільки програмна частина проекту використовує збереження даних авторизації користувача для підстановки, використовується алгоритм шифрування AES. AES є шифром, який може обробляти 128-бітові блок, використовуючи ключі розміром 128, 192 і 256 біт. Серед його переваг можна виділити безпеку, просту реалізацію та легку обробку, що не потребує багато ресурсів [12].

При запуску додатку користувач одразу потрапляє на сцену, де присутня модель «смарт-будинку». На цьому етапі видно елементи User Interface в правому та лівому верхніх кутках, які можна використовувати для відкриття меню телефону для управління або меню паузи відповідно. У лівому нижньому кутку також розташовані Debug елементи, які дозволяють безпосередньо запускати сценарії або емулювати сигнал з датчиків, рис. 3.

Рис. 3. Екран з меню для налаштування датчиків

Після натискання кнопки «меню» користувач бачить, як на екрані з'являється телефон, який симулює додаток для управління «смарт-будинком». Першим кроком є процес авторизації, де користувач може створити новий акаунт або увійти за допомогою наявного. Процес авторизації також надає можливість зберігати дані, щоб не потрібно було вводити їх при кожному вході.

Після успішного проходження процесу авторизації з'являється меню налаштування датчиків, де вгорі відображається ім'я користувача. У цьому меню представлені всі доступні датчики, які можна увімкнути або вимкнути. Після внесення змін до налаштувань, нові дані відправляються на сервер та зберігаються. Таким чином, при наступному відвідуванні користувач побачить збережені налаштування, які використовувались раніше. Він також може випробувати переключення деяких датчиків, щоб побачити вікно підтвердження збереження.

Після налаштування датчиків, користувач може перейти до тестування сценаріїв, які вже реалізовані і виконуються в додатку. Ці сценарії розроблені таким чином, що вони послідовно виконуються один за одним, і новий сценарій не почнеться, поки попередній не буде завершений.

Для початку тестування користувач може запустити сценарій пожежі. Цей сценарій починається з ефекту пожежі, який відтворюється в будинку. Після спалаху пожежі, на місце прибуває пожежна служба та розпочинає гасіння полум'я. Після успішного гасіння, відбувається анімація, що зображує ефект захисту будинку, і пожежна служба від'їжджає від об'єкту, рис. 4.

Інші сценарії працюють за аналогічним принципом. Однак, вони відрізняються ефектами, які відтворюють певні надзвичайні ситуації, а також службами, які прибувають для їх усунення. Щоб протестувати інші сценарії, користувач може натискати кнопки «Water», «Door» та «Humidity».

Рис. 4. Сценарій пожежі та реагування пожежної служби

У системі смарт-будинок інформація з датчиків, які передаються на мікроконтролер Arduino, підлягає процесам шифрування та стиснення з метою забезпечення безпеки та ефективності передачі даних [13].

Шифрування використовується для захисту конфіденційності даних під час їх передачі. Зазвичай використовуються симетричні або асиметричні криптографічні алгоритми. Пропонується шифрування інформації отриманої з датчиків системи «смарт-будинок» за допомогою матричної решітки [14].

Стиснення даних використовується для зменшення обсягу передаваних даних і забезпечення

ефективної передачі по обмежених каналах зв'язку [15]. Для стиснення даних використовуються різні алгоритми стиснення, такі як алгоритми безвтратного стиснення (наприклад, алгоритми Huffman або Lempel-Ziv-Welch) або алгоритми стиснення із втратами. Це дозволяє ефективно використовувати пропускну здатність мережі і зменшує використання ресурсів пам'яті на мікроконтролері.

Таким чином, застосування шифрування та стиснення інформації з датчиків системи смарт-будинок перед їх передачею на мікроконтролер Arduino дозволяє забезпечити безпеку даних, а також ефективне використання ресурсів мережі та пам'яті.

Висновки та перспективи подальшого дослідження. У статті детально розглядаються різні аспекти моделювання системи датчиків, включаючи процес вибору оптимального мікроконтролера, розробку ефективних алгоритмів збору та обробки даних, а також імплементацію шифрування для захисту інформації. Сценарії реагування на порушення безпеки будинку працюють точно і демонструють ефективність системи розумного дому. Крім того, тестування сервісу авторизації користувачів підтверджує, що система зберігає налаштування датчиків для кожного користувача.

Потенційними напрямками покращень у майбутньому є забезпечення бездротового зв'язку між мікроконтролером, датчиками та системою за допомогою Wi-Fi або Bluetooth модулів, що збільшить гнучкість і зручність використання системи. Інший напрямок – покращення та оновлення архітектури модуля авторизації користувачів, щоб забезпечити більш високий рівень безпеки та зручності входу в систему. Ці покращення спрямовані на подальше вдосконалення функціональності та використання системи розумного дому, забезпечуючи більш широкі можливості та поліпшений досвід користувачів.

Список бібліографічного опису

1. Mowad, M. A. E. L., Fathy, A., & Hafez, A. (2014). Smart home automated control system using android application and microcontroller. International Journal of Scientific & Engineering Research, 5(5), 935-939.
2. Kucera, E., Haffner, O., & Kozak, S. (2018). Connection between 3D engine unity and microcontroller arduino: A virtual smart house. Cybernetics & Informatics, 102-109.
3. Kamelia, L., Effendi, M. R., & Pratama, D. F. (2018) Integrated smart house security system using sensors and RFID. In 2018 4th International Conference on Wireless and Telematics, 67-72.
4. Teslyuk, V., Beregovskyi, V., Denysyuk, P., Teslyuk, T., & Lozynskyi, A. (2018). Development and implementation of the technical accident prevention subsystem for the smart home system. International Journal of Intelligent Systems and Applications, 12(1), 13-19.
5. Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017) Smart Home Security System Design Sensor Based on Pir and Microcontroller. International Journal of Global Sustainability, 1(1), 67-73.
6. Pirbhulal, S., Zhang, H., E Alahi, M. E., Ghayvat, H., Mukhopadhyay, S. C., Zhang, Y. T., & Wu, W. (2016). A novel secure IoT-based smart home automation system using a wireless sensor network. Sensors, 17(1), 69.
7. Gaikwad, P. P., Gabhane, J. P., & Golait, S. S. (2015) A survey based on Smart Homes system using Internet-of-Things. In 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), 330-335.
8. Bugeja, J., Jacobsson, A., & Davidsson, P. (2016) On privacy and security challenges in smart connected homes. In 2016 European Intelligence and Security Informatics Conference (EISIC), 172-175.
9. Kumar, P., Gurtov, A., Iinatti, J., Ylianttila, M., & Sain, M. (2015) Lightweight and secure session-key establishment scheme in smart home environments. IEEE Sensors Journal, 16(1), 254-264.
10. Ali, B., & Awad, A. I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. sensors, 18(3), 817.
11. Jacobsson, A., Boldt, M., & Carlsson, B. (2016) A risk analysis of a smart home automation system. Future Generation Computer Systems, 56, 719-733.
12. Adiono, T., Harimurti, S., Manangkalangi, B. A., & Adijarto, W. (2018) Design of smart home mobile application with high security and automatic features. In 2018 3rd international conference on intelligent green building and smart grid (IGBSG), p. 43-47.
13. Yarmilko, A., Rozlomii, I., & Kosenyuk, H. (2021) Hash Method for Information Stream's Safety in Dynamic Cooperative Production System. In International scientific-practical conference. Cham: Springer International Publishing, 173-183.
14. Розломій І.О., Косенюк Г.В., Науменко С.В. (2023) Метод векторного шифрування інформації на основі використання авторського шаблону. Computer-Integrated technologies: education, science, production., (51), 87-93.
15. Розломій І.О. (2022) Метод побудови матричних решіток Кардано для стиснення інформації. Вісник ХНУ. Технічні науки, 1(305), 85-90.

References

1. Mowad, M. A. E. L., Fathy, A., & Hafez, A. (2014). Smart home automated control system using android application and microcontroller. International Journal of Scientific & Engineering Research, 5(5), 935-939.
2. Kucera, E., Haffner, O., & Kozak, S. (2018). Connection between 3D engine unity and microcontroller arduino: A virtual smart house. Cybernetics & Informatics, 102-109.

3. Kamelia, L., Effendi, M. R., & Pratama, D. F. (2018). Integrated smart house security system using sensors and RFID. In 2018 4th International Conference on Wireless and Telematics, 67-72.
4. Teslyuk, V., Beregovskyi, V., Denysyuk, P., Teslyuk, T., & Lozynskyi, A. (2018). Development and implementation of the technical accident prevention subsystem for the smart home system. International Journal of Intelligent Systems and Applications, 12(1), 13-19.
5. Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017). Smart Home Security System Design Sensor Based on Pir and Microcontroller. International Journal of Global Sustainability, 1(1), 67-73.
6. Pirbhulal, S., Zhang, H., E Alahi, M. E., Ghayvat, H., Mukhopadhyay, S. C., Zhang, Y. T., & Wu, W. (2016). A novel secure IoT-based smart home automation system using a wireless sensor network. Sensors, 17(1), 69.
7. Gaikwad, P. P., Gabhane, J. P., & Golait, S. S. (2015, April). A survey based on Smart Homes system using Internet-of-Things. In 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), 330-335.
8. Bugeja, J., Jacobsson, A., & Davidsson, P. (2016, August). On privacy and security challenges in smart connected homes. In 2016 European Intelligence and Security Informatics Conference (EISIC), 172-175.
9. Kumar, P., Gurtov, A., Iinatti, J., Ylianttila, M., & Sain, M. (2015). Lightweight and secure session-key establishment scheme in smart home environments. IEEE Sensors Journal, 16(1), 254-264.
10. Ali, B., & Awad, A. I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. sensors, 18(3), 817.
11. Jacobsson, A., Boldt, M., & Carlsson, B. (2016). A risk analysis of a smart home automation system. Future Generation Computer Systems, 56, 719-733.
12. Adiono, T., Harimurti, S., Manangkalangi, B. A., & Adijarto, W. (2018, April). Design of smart home mobile application with high security and automatic features. In 2018 3rd international conference on intelligent green building and smart grid (IGBSG), p. 43-47.
13. Yarmilko, A., Rozlomii, I., & Kosenyuk, H. (2021, June). Hash Method for Information Stream's Safety in Dynamic Cooperative Production System. In International scientific-practical conference. Cham: Springer International Publishing, 173-183.
14. Rozlomii I.O., Kosenyuk G.V., Naumenko S.V. (2023) A method of vector encryption of information based on the use of an author's template.Computer-Integrated technologies: education, science, production, (51), 87-93.
15. Rozlomii, I.O. (2022) Method of construction matrix Cardano's grids for compression of information. KHNU Bulletin: Technical Sciences, 1(305), 85-90.