Plag

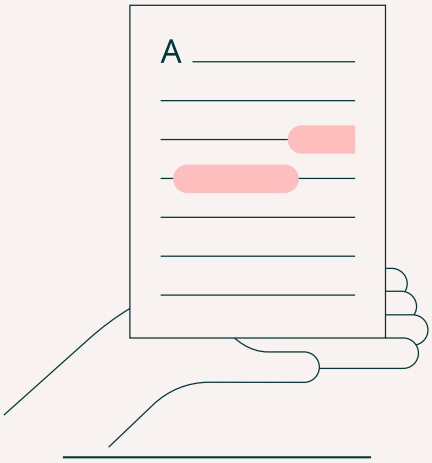
Звіт про оригінальність

● Оцінка схожості 25%

● Ризик плагіату НАЙВИЩИЙ

Ігор Кагало 2025-06-14 10:13

Посилання на звіт: 10bhQ / Посилання користувача: qAHy



Ось вона –
Ваша звіт про оригінальність!

Ми раді повідомити, що перевірка вашого документа завершена, і результати вже готові! Наші алгоритми старанно працювали, щоб знайти збіги в наших базах даних.

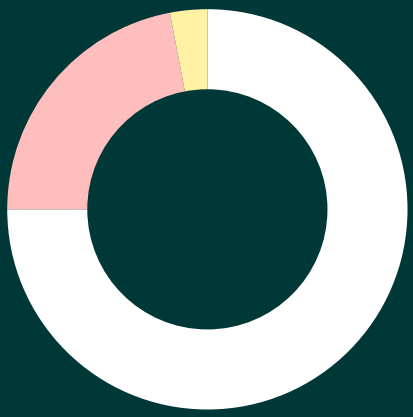
На наступних сторінках ви знайдете результати перевірки:

Бали
Збіги
Посилання

Ваш документ було перевірено за такими джерелами:

- ✓ База даних інтернет-джерел
- ✓ База даних наукових статей
- ○ Глибока перевірка (наш вдосконалений алгоритм)

Бали



Збіги тексту	22%
Перефразування	3%
Цитований текст	0%
Неправильне цитування	0%
Збігів не знайдено	75%

Ризик плагіату НАЙВИЩИЙ

Ризик плагіату вказує, як збіги тексту розподілені по документу. Вищий ризик виникає, коли збіги з'являються близько один до одного, наприклад, у тому самому абзаці або розділі.

Оцінка схожості % 25

Оцінка схожості показує, скільки слів або символів у вашому документі збігаються з текстами інших документів, включаючи перефразовані тексти або неправильні цитати.

Збіги

1 2 ОПИС ОСНОВНИХ ЕЛЕМЕНТІВ ТА ПРИНЦИПИ РОБОТИ СИСТЕМ КОНТРОЛЮ ВІДВІДУВАННЯ

2 1.1 Загальні поняття СКВ

Контроль та облік присутності осіб у різних місцях чи на подіях має давню історію. На кожному етапі існування людства потреба в такому обліку виникала в військових структурах, медицині, політичних колах, освітніх закладах. Весь цей облік традиційно здійснювався на папері. Часто для виконання цієї роботи виділяли окремих працівників. Обробка таких даних, їх сортування та отримання статистичної інформації були досить складними завданнями. Також існував значний ризик втрати інформації, оскільки паперові носії вразливі до фізичних впливів.

З розвитком технологій почали з'являтися цифрові 2 системи контролю відвідуваності. Їхні переваги очевидні. Стало можливим оперувати великими масивами даних, їх систематизувати та структурувати, значно зменшилися ризики їхньої втрати, зникла потреба витрачати час та ресурси, а також ідентифікація великої кількості людей стала можливою за короткий час.

2 Система контролю відвідуваності (СКВ) – це комплекс інструментів та програмного забезпечення, що забезпечує контроль входу, виходу та переміщення осіб в певній будівлі чи на території. СКВ також обробляє інформацію та веде статистику. Найбільш поширеними є системи, засновані на технології RFID.

Особа, що потребує реєстрації в системі, використовує ідентифікатор. Для зчитування даних з мітки використовується зчитувальний прилад (зчитувач). Отриману інформацію обробляє контролер та записує у локальну базу даних або, використовуючи інтернет з'єднання, передає її на сервер. Вся інформація зберігається в базах даних. Це забезпечує швидку обробку та перегляд даних, а також, завдяки використанню резервного копіювання або зберігання на кількох серверах, унеможливлює втрату даних, яка була б характерна для паперових носіїв. Оброблені відомості відображаються в настільній програмі або веб додатку, дозволяючи переглядати статистику, формувати звіти чи виконувати інші операції.

1.2 Історія появи RFID

У 1935 році шотландський науковець Роберт Александер Ватсон-Ватт уперше презентував пристрій для виявлення об’єктів за допомогою радіохвиль — прототип сучасного радара. Незабаром ця технологія була впроваджена як союзниками (зокрема Великою Британією та США), так і Третім Рейхом. Обидві сторони активно експлуатували радіолокаційні системи для виявлення авіації противника.

Проте, тогочасні радари мали низьку точність у визначенні точного місцезнаходження повітряних суден і працювали в межах обмеженого діапазону. Для вирішення цієї проблеми німецькі інженери запропонували цікавий підхід: при поверненні на базу літак змінював траєкторію, що призводило до змін у відбитому радіосигналі, створюючи унікальний візерунок. Цей метод можна вважати аналогом пасивної RFID-системи, оскільки він дозволяв розпізнавати об'єкти за характером відбитого сигналу.

Незважаючи на інноваційність такого методу, він вимагав від пілотів значної концентрації та зусиль. Тому Ватсон-Ватт розробив концепцію автоматизованої ідентифікації літаків — систему розпізнавання "свій-чужий". У британських винищувачах встановлювали спеціальні передавачі, які отримували радіосигнал із землі й у відповідь транслювали код, що підтверджував "дружню" приналежність повітряного судна. Такий принцип став основою сучасної RFID-технології, де взаємодія між зчитувачем і передавачем дозволяє ідентифікувати об’єкти.

Від 1966 року технологія RFID почала використовуватися в США, Європі та Японії для запобігання крадіжкам 6 у роздрібній торгівлі. У 1973 році Маріо В. Кардулло отримав патент на перший активний RFID-тег з можливістю повторного запису. У тому ж році Чарльз Волтон запатентував пасивний варіант мітки.

Ці винаходи стали базою для сучасних систем контролю доступу, зокрема — домофонів. Для демонстрації роботи RFID Волтон вставив мітку в пластикову карту та розмістив зчитувач біля дверей. Коли карта наближалась до пристрою, відбувалося зчитування й ідентифікація, після чого двері автоматично відкривались.

Згодом RFID почали застосовувати в державному секторі, зокрема для моніторингу ядерних матеріалів на об’єктах у Лос-Аламосі. Передавач кріпився до вантажного транспорту, а зчитувач — до контрольно-пропускного пункту. У разі збігу коду передавача з даними зчитувача, система відкривала ворота. Аналогічні рішення стали основою для автоматизованих систем оплати на платних дорогах.

У 1987 році запрацювала перша автомагістраль, яка використовувала RFID-технологію для збору плати за проїзд. Ці рішення залишаються актуальними й сьогодні.

До 2000 року було зареєстровано понад тисячу патентів, пов’язаних з RFID. У 2015 році

обсяг цього ринку оцінювався вже в 26 6 мільярдів доларів США, тоді як у 2005 він становив лише близько 2 мільярдів. Таким чином, за одне десятиліття відбувся стрімкий ріст на понад 24 мільярди. Аналітики прогнозують, що технологія RFID і надалі набуватиме популярності, щонайменше протягом наступних двох десятиліть. Вона вже сьогодні є невід'ємною частиною багатьох галузей.

1.3 Принципи роботи RFID

Системи радіочастотної ідентифікації (RFID) не базуються на єдиному універсальному стандарті чи технології; кожна з них використовує свій власний набір стандартів. 2 Проте, метод передачі інформації залишається незмінним для всіх – радіохвиля. Основними складовими RFID-системи виступають: 2 зчитувач, антена та мітка. Зрозуміло, це тільки базова конструкція, що розширюється програмним забезпеченням зчитувачів, потрібними драйверами, а також алгоритмами 2 кодування та ідентифікації.

2 Радіочастотна ідентифікація, по суті, виконує ті самі функції, що й штрих-коди або магнітні смуги. 2 Але ця технологія відносно нова та має значні переваги. Їй не потрібен прямий візуальний контакт між компонентами, вона працює на відстані та володіє значно більшою стійкістю до зовнішніх впливів.

На рисунку 1.1 наведено приклад роботи RFID.

2 Рисунок 1.1 — Приклад роботи 1 RFID

1 Технологія RFID звільняє від потреби прямого контакту чи візуального контролю 1 між міткою та сканером, надаючи можливість зчитувати багато міток одночасно в межах радіуса дії антени. RFID-системи також зберігають функціональність у складних умовах, оскільки RFID-мітки стійкі до зовнішніх впливів, завдяки радіопрозорому корпусу. Обмін інформацією між сканером та міткою відбувається безперешкодно, ігноруючи бруд, 1 фарбу, пару, воду, пластик та дерево.

1 Мітка складається з інтегральної схеми, де 1 зберігається інформація, 1 та антени, яка забезпечує радіозв'язок. Існують мітки з різними типами живлення.

Вони поділяються на активні, пасивні та напівпасивні [6]. Активні мітки забезпечені власним джерелом енергії, 5 що робить їх придатними для використання в екстремальних умовах. Пасивні мітки не мають власного живлення, а 1 отримують енергію від зчитувача через магнітне або електричне поле. Найчастіше їх використовують для ідентифікації. Напівпасивні мітки 1 працюють від власного джерела живлення, але використовують його лише для активації мікросхеми, а не для обміну даними зі сканером. Детальний порівняльний аналіз представлений в таблиці

1.1.

1 Таблиця 1.1 — Порівняння активних та пасивних міток

1 Критерій

1 Активний тег

1 Пасивний тег

1 Батарея

1 Є

1 Нема

1 Джерело живлення

1 Вмонтоване в мітку

1 Живиться від зчитувача

1 Наявність живлення

1 Постійне

1 Тільки в полі дії зчитувача

1 Необхідна потужність 1 зчитувача

1 Низька, потрібна лише для перенесення 1 даних

1 Висока, мусить живити 1 тег

1 Сила сигналу від мітки до 1 зчитувача

Висока

1 Низька

1 Діапазон дії

1 Великий, більше 100 метрів

Короткий або дуже короткий, 1 3 метри і менше

1 Термін служби тега

1 Обмежений зарядом батареї 1 (залежить 1 від типу енергозбереження)

1 Дуже 1 довгий

1 Фізичний розмір

1 Великий

1 Малий

1 Одночасне зчитування міток

1 Зчитує 1000 тегів на площі 28000 м². 2 20 тегів, які рухаються зі швидкістю понад 11 160 11 км/год

2 Зчитує 100 тегів в межах 100 2 м 2 від зчитувача. 20 тегів, які рухаються зі швидкістю менше 8 км/год

Передача даних із сенсорів

2 Можливість безперервного контролю давача. Записується час подій

2 Зчитує і передає дані лише при живленні від зчитувача. Час подій не записується

2 Пам’ять

2 Велика кількість пам’яті (вимірюється в кілобайтах), є можливість пошуку

2 Невеликий 2 обсяг 2 (вимірюється в байтах)

2 Типове застосування

2 Динамічні бізнес-процеси: 2 безпека, зондування, 2 збереження даних, моніторинг залізничних вагонів, вантажів 2 тощо

2 Жорсткі бізнес-процеси: 2 маркування предметів, багажу, коробок, етикеток, контроль доступу, трекер відвідуваності

2 Ціна

2 Висока — від 5$ до 100$

2 Низька

2 На рисунку 1.2 продемонстровано будову RFID тега.

Рисунок 1.2 — Будова RFID тега

Теги розрізняються також за частотним діапазоном. Існують НЧ, низькочастотні (біля 130 кГц), ВЧ, високочастотні (до 13 МГц), і УВЧ, ультрависокочастотні (приблизно 900 МГц), що можна побачити на рисунку 1.3. Найбільш поширені високочастотні теги. Їх застосовують у банківських системах, в системах ідентифікації та контролю доступу, на транспорті та в торгівлі. Проте їх неможливо зчитати на великій відстані або за несприятливих погодних умов. Певною мірою цю проблему вирішують ультрависокочастотні мітки [6].

RFID мітки використовують три типи пам’яті: RO, WORM, RW.

RO (read only, лише читання) – під час виробництва в них записують дані лише один раз. Вони стійкі до підробки чи перезапису. Мають унікальний ідентифікаційний номер, який використовується для ідентифікації.

WORM (write once read many, один раз записати, багато разів прочитати) – аналогічно RO, мають унікальний id, але також мають блок пам’яті, куди можна одноразово записати інформацію. Надалі перезапис неможливий, проте є можливість багатократно зчитувати дані.

RW (read and write, читати та писати) – крім ідентифікатора мають блок пам’яті. Його можна перезаписувати та зчитувати безліч разів.

На рисунку 1.3 можна побачити приклади пасивних міток.

Рисунок 1.3 — Три приклади пасивних міток

Залежно від способу виробництва, мітки можуть бути різноманітними. Їх створюють у вигляді пластикових карток, брелоків з пластику, наклейок або навіть капсул, призначених для імплантації під шкіру.

Існує великий вибір різних зчитувачів. Основним завданням зчитувачів є розпізнавання унікального ідентифікатора (ID), а також інформації, збереженої в пам'яті мітки. Ще однією їх функцією є можливість запису даних до пам'яті мітки. Залежно від дистанції зчитування, зчитувачі класифікують на такі типи: близької дії, середньої (до 50 см) та дальньої (понад метр). Зчитувачі бувають стаціонарними або портативними.

Оскільки сучасна RFID-технологія є досить доступною, надійною та простою у використанні, її найчастіше використовують при розробці систем обліку відвідуваності чи контролю доступу.

1.4. Застосування технології RFID для ідентифікації та доступу

Безконтактні технології, зокрема, на базі RFID, набули широкого розповсюдження для керування доступом та ідентифікації осіб. Яскравий приклад - готельний бізнес, де RFID-карти використовуються для організації контролю доступу до конкретних номерів або зон. Зазвичай, це забезпечує гостю можливість потрапити у замовлений номер, а за потреби – також до зон відпочинку або інших об'єктів з обмеженим доступом. Крім того, RFID-технології дають змогу відслідковувати кількість відвідувачів готелю або окремих його частин, що сприяє ефективному моніторингу та аналізу отриманої інформації.

В історичному розрізі для ідентифікації застосовувалися низькочастотні RFID-мітки. Вони функціонували разом зі 4 спеціальним програмним забезпеченням, яке в реальному часі давало змогу зчитувачам, інтегрованим із системою, зчитувати інформацію з тегів та перевіряти її на сервері або контролері для підтвердження особи та надання дозволу на доступ.

Останніми роками спостерігається значна модернізація інфраструктури - галузь поступово переходить від підтримки тільки низькочастотного обладнання до сучасних високочастотних RFID-пристроїв. 4 Це відкриває нові можливості функціоналу систем: крім звичного контролю доступу, стає можливою розробка цифрових перепусток, ідентифікація особи, здійснення електронних платежів за проживання та інших зручностей.

З-поміж компаній, що активно впроваджують RFID-рішення, особливо вирізняється Mifare, яка розробила один із найпопулярніших стандартів у цій сфері.

Окремо варто зазначити використання RFID-технологій 8 в автомобільній промисловості. Сучасні транспортні засоби можуть бути обладнані відповідними зчитувачами: приклавши смартфон з підтримкою RFID до визначеної зони, користувач має змогу відімкнути двері або навіть запустити двигун. Це розширює можливості безконтактного керування та підвищує комфорт користування.

2 2 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ КОНТРОЛЮ ВІДВІДУВАНОСТІ

2 2.1. Компоненти СКВ

2 На вітчизняному 2 та світовому ринку представлено розмаїття 2 готових систем контролю відвідуваності, або ж систем контролю та керування доступом. Їхніми рішеннями послуговуються чисельні державні структури, представники бізнесу та заклади освіти. Найбільш простим 2 прикладом може бути домофон, котрий встановлений майже в кожному багатоповерховому будинку 2 біля вхідних дверей. Він теж використовує технологію радіочастотної ідентифікації.

Переважна більшість СКВ для взаємодії з користувачем застосовує RFID технології. Існують як масштабні системи, що охоплюють цілі будівлі чи мережу закладів та мають багато функцій, так і невеликі пристрої, що використовуються безпосередньо на одному чи кількох входах (згаданий вище домофон). Для створення різних систем застосовуються багато різноманітних компонентів. Від вибору цих компонентів напряму залежить вартість системи, її функціональність, зручність використання, здатність до розширення, вдосконалення, надійність та безпека.

Рисунок 2.1 демонструє комплектацію бюджетної СКВ.

Рисунок 2.1 — Комплектація бюджетної СКВ.

До основних компонентів апаратної частини СКВ входять: мікроконтролер, пристрій для зчитування міток, механізм оповіщення (дисплей, гучномовець або світловий індикатор). Щодо програмного забезпечення, виокремлюємо: прошивку контролера, базу даних (місцеву або віддалену), веб-додаток або мобільний клієнт (не завжди обов'язково). На сучасному ринку електроніки доступний великий вибір цих складників. Це дозволяє підібрати їх так, щоб отримати оптимальне співвідношення ціни та якості. У цьому розділі розглянуто підбір компонентів для розробки власної системи контролю відвідуваності студентів, з орієнтацією на мінімальну вартість та забезпечення основних функціональних можливостей.

2.2. Вибір мови програмування для розробки плати та веб застосунку

У процесі створення системи контролю відвідуваності (СКВ) важливим етапом є правильний вибір програмного забезпечення, оскільки він охоплює два основні компоненти. Перший — це написання прошивки для апаратної частини, де необхідна мова програмування, що дозволяє ефективно взаємодіяти з електронними компонентами. Другий — розробка веб-застосунку, який працюватиме на сервері та взаємодіятиме з базою даних.

Рішення щодо програмних інструментів суттєво впливає на продуктивність розробки, гнучкість системи та легкість її підтримки. Найпопулярнішими мовами сьогодні є Python, Java, JavaScript, C++, C#, PHP — вони широко застосовуються у створенні як серверних, так і десктопних та веб-програм. Для роботи з мікроконтролерами традиційно використовують низькорівневі мови на кшталт C/C++ або Assembler, адже вони забезпечують прямий доступ до апаратного забезпечення. Проте розробка на них часто супроводжується складнощами — програмістам доводиться багато часу витрачати на пошук та виправлення помилок, що виникають через технічну складність мов та необхідність глибоких знань апаратного рівня.

У зв’язку зі зростанням обчислювальних можливостей контролерів та плат, почали

з’являтися нові інструменти, які можуть слугувати альтернативою традиційним C/C++. Серед них — MicroPython (спрощена версія Python для мікроконтролерів) і Espruino (JavaScript-фреймворк для мікроконтролерів). Незважаючи на меншу продуктивність, вони значно полегшують написання коду завдяки використанню високорівневих мов. У проектах, де не критична швидкість виконання, ці варіанти цілком доцільні.

Для реалізації програмної частини системи контролю відвідуваності студентів було обрано Python. Це високорівнева, інтерпретована мова з динамічною типізацією, яка підтримує кілька парадигм програмування: імперативну, функціональну та об'єктно-орієнтовану [4]. Вона дозволяє розробнику сконцентруватися на вирішенні прикладних задач, завдяки простому і зрозумілому синтаксису.

Однією з переваг Python є невелика крива навчання — її легко опанувати навіть початківцям. Крім того, велика кількість бібліотек і модулів забезпечує розширені можливості — від роботи з файлами та мережами до обробки тексту, тестування, інтеграції з ОС тощо. Python активно використовується у різних галузях: для написання сценаріїв, веб-розробки, автоматизації, машинного навчання, створення прикладних програм тощо. У разі потреби підвищення швидкодії, можна вбудовувати фрагменти коду, написані мовами C або C++.

Мова є кросплатформною, отже її застосування можливе практично на будь-якій операційній системі. Основним недоліком Python залишається його відносно низька швидкість виконання. Проте завдяки зручності розробки та скороченню загального часу створення програмного продукту, цей недолік часто не є критичним.

Згідно з аналітичними звітами, Python стабільно посідає провідні позиції у рейтингах популярності мов програмування та продовжує стрімко розвиватися. Враховуючи універсальність, простоту та широкі можливості, Python є оптимальним вибором для реалізації невеликої, але функціональної системи контролю відвідуваності.

Для організації взаємодії з апаратною частиною системи було обрано MicroPython — легку версію мови Python, спеціально адаптовану для мікроконтролерів. Ця платформа побудована на основі синтаксису Python 3 і реалізована мовою C, що забезпечує її оптимізацію для середовищ з обмеженими обчислювальними ресурсами. Основна її відмінність від стандартного Python полягає у спрощеній структурі та скороченому наборі вбудованих модулів. Повноцінна стандартна бібліотека недоступна, однак присутній мінімально необхідний функціонал, якого зазвичай вистачає для виконання базових завдань.

Замість традиційних бібліотек, MicroPython пропонує спеціалізовані модулі для прямого доступу до апаратних компонентів. Зокрема, мова підтримує роботу з GPIO — портами загального призначення, які дозволяють контролеру взаємодіяти з різноманітними

периферійними пристроями. Крім того, у випадку використання плат із підтримкою Wi-Fi, доступні бібліотеки для реалізації мережевих функцій [8].

Завдяки MicroPython створення програм для керування електронікою стало значно простішим і доступнішим, особливо для новачків у цій галузі. Його основна мета — спростити процес розробки вбудованих систем настільки, щоб ним могли користуватись як професіонали, так і ентузіасти. Він активно застосовується у навчальних проєктах, дослідженнях, прототипуванні домашніх систем, а подекуди — й у комерційних розробках.

Однією з найбільш зручних особливостей MicroPython є наявність інтерактивного середовища REPL (Read-Eval-Print Loop), яке дозволяє взаємодіяти з мікроконтролером у реальному часі. Через REPL можна подавати команди без попередньої компіляції або завантаження повного скрипта в пам’ять пристрою. Для роботи можна використовувати зручні середовища розробки, як-от Thonny або uPyCraft, а також звичайний термінал чи спеціальні веб-інтерфейси.

У структурі проєкту важливими є два ключові файли. Перший — boot.py, який виконується під час кожного запуску пристрою. У ньому зазвичай зберігають налаштування й конфігураційні параметри. Другий — main.py, в якому розміщується основна логіка програми. Він автоматично запускається після завершення виконання boot.py.

Для розробки веб-застосунку системи відвідуваності було застосовано Django. Цей фреймворк, як і MicroPython, створений для мови Python. Він є безкоштовним та з відкритим вихідним кодом, його розроблено з метою пришвидшення створення вебсайтів та зменшення витрат часу, ґрунтуючись на шаблоні MVC [1]. Під час розробки вебсайтів часто зустрічаються однотипні операції, саме для їх вирішення й розробляються веб-фреймворки. Django з'явився у 2005 році для розв'язання цих труднощів, і з того часу він став одним із найбільш затребуваних інструментів.

Він пропонує все необхідне для оперативного та зрозумілого кодування. Ці інструменти спрощують роботу з аутентифікацією, адмініструванням, шаблонами та безпекою. Розроблені вебсайти без зусиль масштабуються та здатні обробляти значні обсяги трафіку. Вебсайт побудовано з кількох додатків, які слід відокремлювати.

Для створення бази даних модель описується через класи, на основі яких формується схема бази. Забезпечено сумісність з усіма 10 популярними системами керування базами даних. Немає потреби витрачати час на розробку адміністративної панелі, оскільки можна використати готову, яка за замовчуванням вже є у фреймворку. Django Rest Framework — це практичний інструмент для створення API. Він працює зі

стандартними моделями та дозволяє створити зрозумілий API-інтерфейс для будь-якого проєкту. Це надзвичайно важливо, адже контролер взаємодіятиме з сервером саме через API.

Підсумовуючи, Django ідеально підходить для розробки СКВ. Він надає все необхідне, процес розробки повинен бути відносно простим, а система може бути досить гнучкою.

2.3. Вибір мікроконтролера

Мікроконтролер — це мікропроцесорна система, що зібрана в одному кристалі. Він містить в собі один або декілька процесорів, пам'ять, а також різноманітні периферійні пристрої для введення і виведення даних. Мікроконтролери використовуються для керування вбудованими системами, або іншими електронними пристроями. Вони є основою для систем контролю доступу, управління двигунами, медичного обладнання, іграшок, пультів дистанційного керування, а також для побутових приладів. Широко використовуються у сфері Інтернету речей (IoT) та в багатьох інших різноманітних приладах. Їх розміри та вартість значно менші, ніж у пристроїв, що використовують окремий мікропроцесор. Існує великий вибір контролерів з різними характеристиками. Вони поділяються на три основні категорії: 8-розрядні, 16- та 32-розрядні, та цифрові сигнальні процесори [5]. Для зменшення енергоспоживання, деякі моделі можуть працювати на низьких частотах. Зазвичай, вони зберігають свою функціональність, очікуючи ті чи інші події. Споживання енергії в режимі сну вкрай низьке, що надзвичайно корисно для систем, де важливим є тривале збереження заряду батареї.

Під час проєктування, зокрема не лише СКВ, а й будь-якої іншої автоматизованої системи, вибір мікроконтролера є одним з найважливіших етапів. Значну популярність здобули так звані плати розробника. Вони містять в собі контролер, а також інші компоненти, як, наприклад, антена Wi-Fi чи GSM, GPS модуль тощо. Таке рішення дозволяє значно зекономити час та кошти на розробку власної плати. Готові рішення підходять для розв'язання більшості задач. Найбільш популярні плати на основі мікроконтролерів ESP, AVR та STM. Raspberry Pi також широко застосовується. Вона дорожча, але її функціональні можливості значно ширші.

Для розробки нескладної студентської СКВ буде цілком достатньо недорогої плати, яка може взаємодіяти зі зчитувачем та зберігати дані у власній пам'яті або надсилати їх на сервер. Оскільки система буде взаємодіяти із сервером, необхідно обрати плату, з якою буде просто реалізувати такий зв'язок. Не менш важливим параметром є простота взаємодії з платою, тобто складність її підключення та прошивки має бути мінімальною.

Відштовхуючись від усіх вищезгаданих параметрів, було прийнято рішення використовувати для роботи плату NodeMCU Lolin третьої версії. Це недорога платформа з відкритим кодом для IoT. Спочатку вона включала програмне забезпечення, яке працювало на ESP8266 Wi-Fi системі на чипі від Espressif Systems, а також апаратне забезпечення, що базувалося на модулі ESP-12 (рисунок 2.2).

Рисунок 2.2 – WiFi-модуль ESP8266

Згодом була додана підтримка 32-бітного ESP32. Рішення також було обумовлене тим, що для цієї плати є портована прошивка MicroPython та різні бібліотеки для периферійних пристроїв, які можна безкоштовно використовувати.

NodeMCU – це програмне забезпечення з відкритим кодом, навколо якого сформувалося безліч відкритих проєктів, призначених для прототипування. Назва складається з "Node" (вузол) та "MCU" (мікроконтролер). Переважно це стосується саме прошивки, а не певного набору для розробки, що з нею працює. І прошивка, і проекти плат мають відкритий вихідний код.

Сама прошивка базується на мові скриптів Lua. Вона веде свій початок від проєкту eLua та побудована на основі Espressif Non-OS SDK для ESP8266. Всередині використовується багато відкритих компонентів, як от lua-cjson та SPIFFS. Враховуючи обмежені ресурси, користувачі мають можливість вибирати необхідні модулі та формувати програмне забезпечення, приймаючи до уваги ці обмеження. Також реалізована підтримка 32-бітного ESP32.

Зазвичай, апаратна частина для прототипування представляє собою друковану плату в форматі подвійного вбудованого пакета (DIP), яка поєднує USB контролер з компактною платою, на якій розміщені контролер та антена. Формат DIP забезпечує просте прототипування на макетних платах. На самому початку проєкт базувався на модулі ESP-12 ESP8266, який є Wi-Fi SoC, що інтегрований з ядром Tensilica Xtensa LX106 та активно застосовується у розробках інтернету речей.

NodeMCU з'явився невдовзі після виходу ESP8266. 30 грудня 2013 року Espressif Systems випустила ESP8266. Проєкт NodeMCU офіційно стартував 13 жовтня 2014 року, коли Хонг додав перший файл прошивки nodemcu на GitHub. Через два місяці проєкт поширився, охопивши відкриті апаратні платформи, після того як розробник Хуанг Р. створив гербер-файл для ESP8266 під назвою DevKit v0.9. Трохи згодом Туан РМ переніс клієнтську бібліотеку MQTT з Contiki на платформу ESP8266 та зробив коміт у проєкті, що додало платі підтримку протоколу IoT MQTT, використовуючи Lua для доступу до MQTT. Ще одним значущим оновленням було впровадження 30 січня 2015 року, коли Девсаурус переніс u8glib у проєкт, що дало

змогу платі з легкістю керувати LCD, OLED та навіть VGA-дисплеями. Влітку 2015 року засновники покинули проєкт, а подальший розвиток платформи перейняла група незалежних розробників. До літа 2016 року NodeMCU вже налічував понад 40 різних модулів.

Плата проста в експлуатації завдяки зручному підключенню до Wi-Fi. Вона підтримує стандарт WiFi 802.11 b/g/n. Живлення вимагає 4,5–9 В, підключення відбувається через USB. Під час обміну даними споживає приблизно 70 мА, у режимі очікування – менше ніж 200 мкА. Можна стверджувати, що плата є досить енергоефективною. Підтримуються інтерфейси UART та GPIO. Прошивка здійснюється з комп'ютера через USB або через WEBRepl, шляхом Wi-Fi з'єднання. Це зручно, бо додаткові програматори не потрібні.

2.4. Вибір зчитувача

При виборі зчитувача міток необхідно враховувати перелік протоколів, які він здатний підтримувати, швидкість передачі даних, рівень робочої напруги, тип інтерфейсу для інтеграції з мікроконтролером та його габарити. Якщо передбачається експлуатація пристрою поза приміщенням, необхідно звернути увагу на його стійкість до впливу атмосферних явищ.

Серед найпопулярніших та бюджетних рішень можна виділити PN532 та RFID-RC522. У процесі розробки було прийнято рішення зупинитися на останньому, враховуючи його доступнішу ціну та відповідність необхідним параметрам для реалізації СКУД. Даний модуль працює на частоті 13.56 МГц. Завдяки цьому забезпечується розпізнавання як NFC, так і RFID міток. Його часто використовують студенти, радіоаматори та розробники комерційних проектів [2]. Застосування цього модуля дає змогу отримати доступ до всіх необхідних безконтактних функцій. Підтримувані інтерфейси: SPI, I2C та UART. Вибір потрібного інтерфейсу здійснюється налаштуванням логічного рівня на відповідних виводах мікросхеми. Для живлення потрібна напруга 3.3 В. Зчитування міток відбувається на відстані до 6 сантиметрів. Пристрій дозволяє як зчитувати, так і записувати інформацію. Максимальна швидкість передачі даних складає приблизно 10 мегабіт за секунду. На рисунку 2.4 показано RFID модуль RC522.

Рисунок 2.4 — RFID модуль RC522

Існують готові бібліотеки для взаємодії з цим модулем. Серед недоліків слід зазначити відсутність підтримки NFC смартфонів; модуль працює лише з мітками. Однак, для реалізації даної системи контролю доступу цього буде цілком достатньо.

2.5. Вибір модуля індикації

В будь-якій системі керування доступом необхідно закласти певний механізм

сповіщення. Користувач має мати змогу збагнути, чи вдало відбулася ідентифікація, та бачити її наслідок. З цією метою можна використовувати світлові або звукові індикатори, або ж виводити інформацію на екран. Останній варіант найзручніший, оскільки дає змогу відображати будь-які потрібні повідомлення. Це суттєво розширює функціональність і поліпшує комфорт використання системи.

Для реалізації індикації обрано OLED I2C 128x32 дисплей, представлений на рисунку 2.4.

Рисунок 2.4 — OLED дисплей

Як вже зрозуміло з назви, він функціонує за протоколом I2C і має компактні розміри, яких повністю вистачає для відображення ключової інформації. Завдяки OLED технології, цей екран досить економічний у споживанні електроенергії. Його часто використовують при розробці невеликих пристроїв. Головними його перевагами є яскравість, чітке відображення навіть при яскравому освітленні, та безсумнівно, ціна.

3 2 РОЗРОБЛЕННЯ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СКВ

2 3.1. З’єднання мікроконтролера із периферійними пристроями

2 Безконтактна система реєстрації відвідувачів буде втілена на платі, серцем якої стане мікроконтролер ESP8266, інтегрований в NodeMCU v3. Для функціонування системи необхідні зовнішні компоненти: зчитувач міток та екран для відображення інформації. У попередньому розділі було проаналізовано та обґрунтовано використання RFID-RC522 зчитувача та OLED дисплея. Живлення плати забезпечується від джерела 5В через microUSB роз'єм, а компоненти отримують живлення від плати, використовуючи виводи з напругою 3.3 вольта. Загальна схема взаємодії периферійних пристроїв з контролером наведена на рисунку 3.1.

Рисунок 3.1 — Схема з’єднання контролера з периферією

Зчитувач з'єднується через інтерфейс SPI. Такий вибір зумовлено простотою реалізації, економічністю та можливістю швидко передавати інформацію. Шина працює синхронно, використовує чотири дроти. Підключення відбувається за моделлю "ведучий-ведений", де ініціює синхросигнали виключно ведучий пристрій [3]. У цій конфігурації завжди один головний пристрій, проте кількість підлеглих може змінюватись. Взаємодія відбувається виключно у режимі повного дуплексу [3]. Пересилання даних реалізовано з використанням наступних сигналів:

MOSI - для пересилання даних від головного пристрою до підлеглого.

MISO - для передавання даних від підлеглого 1 пристрою до головного.

1 SCK - для 1 передавання тактового сигналу до підлеглих 1 пристроїв.

1 SS - відповідальний за запуск та завершення сесії зв'язку.

RC-522 також обладнано виводом IRQ, що відповідає за переривання (у даному 1 випадку він не використовується), GND - заземлення, Vcc – живлення, і RST, який потрібен для скидання. Розташування контактів продемонстровано 1 на рисунку 3.2.

1 Рисунок 3.2 – Призначення виводів на модулі MFRC-522

1 Розташування 1 контактів на платі показано 1 на рисунку 3.3. На виходах від D1 до D8 програмно можна 1 задати тип сигналу. Однак, для найбільшої швидкодії при взаємодії з SPI інтерфейсом, краще застосовувати 1 виводи D5 — D8, оскільки 1 на 1 них вже на рівні 1 "заліза" зафіксовані 1 сигнали SPI [2]. Їхнє використання звільняє від потреби ініціалізувати невикористані виводи. Схема з'єднання модуля з контролером представлена в таблиці.

1 Рисунок 3.3 — Позначення виводів плати NodeMCU V3

1 Таблиця 3.1 — Підключення зчитувача до плати

1 Позначення виводу на модулі

1 Позначення виводу на платі

1 SDA

1 D8

SCK

D5

MOSI

D7

MISO

D6

IRQ

-

GND

G

RST

D4

3.3 V

3 V

1 Дисплей буде взаємодіяти з платою, використовуючи шину I2C. Інформація передається через кабель даних та кабель тактування. Існують головний пристрій та підлеглий, де головний створює тактові імпульси. Швидкість передачі є значно нижчою, ніж у випадку з SPI. Екран обладнаний наступними виводами: GND – маса, 1 VCC – живлення, SCL – лінія послідовного тактування, SDA – лінія 1 послідовних 1 даних [3]. Схема з'єднання зображена на рисунку 3.4. У таблиці 3.2 представлені варіанти з'єднання контактів екрану з платою.

1 Рисунок 3.4 — Схема з’єднання OLED дисплея із мікроконтролером

1 Таблиця 3.2 — Підключення портів дисплея до портів мікроконтролера

1 Позначення виводу на дисплеї

1 Позначення виводу на контролері

SCL

D1

SDA

D2

1 GND

1 GND

1 VCC

1 3.3V

1 3.2. Програмування мікроконтролера

1 Для забезпечення комунікації між контролером і сервером в системі використовується бібліотека urequests. Ця бібліотека дозволяє робити HTTP-запити з

мікроконтролерів, що є особливо корисним для відправлення даних за допомогою API, наприклад, для зчитування та реєстрації RFID-міток. Кожен запит, який робить контролер, передає ідентифікатор мітки (ID) на сервер для подальшої обробки.

1. Режим сканування. Коли користувач вибирає режим сканування, контролер передає на сервер ідентифікатор мітки (ID). Сервер перевіряє наявність цього ID в системі; якщо ID не знайдено в базі даних, сервер надсилає сповіщення про помилку, інформуючи користувача, що мітка не зареєстрована; якщо ID знайдено, сервер фіксує час та дату сканування, 1 а також ім'я студента, який здійснив зчитування мітки. Ця інформація надсилається назад на контролер, де вона відображається на екрані, підтверджуючи успішну операцію.

2. Режим реєстрації: у випадку вибору режиму реєстрації, контролер також передає ID мітки на сервер. Сервер здійснює наступну перевірку: якщо мітка вже зареєстрована (ID присутнє в базі даних), сервер надсилає повідомлення про дублікат, повідомляючи, що ця мітка вже використовується; якщо ID відсутнє, сервер фіксує час і дату зчитування мітки та її 1 номер. Після цього мітка стає доступною для прив'язки до студента, що дозволяє використовувати її для подальших операцій.

Щоб забезпечити автоматичний запуск системи та її роботу без необхідності вручну запускати кожен раз потрібний режим, у систему можна внести налаштування в файл boot.py. Цей файл відповідає за початкові налаштування і виконання команд одразу після запуску контролера.

Автоматичний запуск режиму: У boot.py записуються інструкції, які налаштовують систему на автоматичний запуск у вибраному режимі (реєстрація чи сканування).

Обмеження функціональності: Така реалізація обмежує роботу системи 1 в одному режимі, тобто 1 контролер може працювати або в 1 режимі реєстрації міток, або в режимі сканування, але не в обох одночасно. Для зміни режиму роботи необхідно редагувати файл boot.py та перезапустити систему.

Цей підхід дозволяє гнучко налаштовувати роботу пристроїв залежно від ролі в системі:

• Адміністративні пристрої для реєстрації: Один або кілька контролерів можуть бути налаштовані для роботи лише в режимі 1 реєстрації нових міток. Це дозволяє адміністрації швидко додавати нові мітки в систему без зміни налаштувань інших пристроїв.

• Пристрої для сканування: Інші контролери, які використовуються для сканування міток студентів, можуть бути налаштовані тільки на режим сканування. Це забезпечує стабільність роботи системи та мінімізує помилки, які можуть виникнути в разі

змішування режимів.

7 Крім того, такий підхід дає змогу швидко змінювати режим роботи на будь-якому контролері в разі необхідності. Для цього достатньо просто змінити налаштування в файлі boot.py та перезапустити контролер. Це зручно, коли потрібно тимчасово змінити функціональність, наприклад, для тестування або в разі заміни одного з пристроїв.

1 3.3. Розробка веб додатку

1 Розроблена система контролю відвідуваності (СКВ) передбачає обмін інформацією з віддаленим сервером, що потребувало створення веб-інтерфейсу, API та адміністративної панелі. Для реалізації цих компонентів було обрано фреймворк Django, який забезпечує швидку розробку веб-застосунків з чіткою структурою.

Оскільки система не передбачає значних навантажень, як систему керування базами даних було обрано SQLite — вбудовану реляційну СУБД, яка інтегрується з Django за замовчуванням і не потребує додаткової конфігурації сервера. Вона ідеально підходить для невеликих або прототипних проєктів, де не використовуються складні запити чи великі обсяги даних.

На першому етапі було визначено структуру бази даних через файл models.py [7]. Архітектура БД включає чотири основні таблиці з різними типами зв'язків: 1 Групи, Студенти, Просканові мітки, а 1 також 1 Мітки для реєстрації. У майбутньому, за потреби розширення функціоналу, можливе додавання таблиць з інформацією про викладачів, аудиторії, факультети, розклад занять тощо. Django легко масштабуються, що дозволяє ефективно адаптувати систему до зростаючих потреб.

Для адміністрування було налаштовано адмін-панель, яка є однією з вбудованих можливостей Django і забезпечує зручний інтерфейс керування даними. Комунікація між апаратною частиною та сервером організована через REST API, реалізований за допомогою бібліотеки 1 Django REST Framework [7].

1 Для серіалізації даних, тобто перетворення їх у зручні для передачі формати, використовуються спеціальні класи-серіалізатори. API працює за двома основними маршрутами, відповідно до режимів роботи СКВ:

Сканування міток — https://tracker.senabo.site/api/scan/

Реєстрація міток — https://tracker.senabo.site/api/register/

Типовий запит до сервера має такий вигляд:

{

"body": {

"tag": "ax673478",

"student": "",

"scanned": null

}

}

При надходженні запиту обробка виконується у файлі views.py, де визначаються дії залежно від типу запиту (GET або POST). У відповідь сервер формує і повертає необхідні дані з БД, які потім використовуються на мікроконтролері.

Для зручного доступу до інформації про 1 відвідування було створено користувацький 1 веб-інтерфейс 1 за адресою https://tracker.senabo.site/. Користувачі можуть переглядати як загальну статистику по академічних групах, так і деталізовані дані по кожному студенту.

Розгортання системи здійснено на віртуальному приватному сервері (VPS), наданому компанією Linode. Серверна частина працює на Ubuntu 18.04, з використанням Nginx як веб-сервера та uWSGI як шлюзу між веб-сервером і Django-додатком. Було також налаштовано систему імен DNS для забезпечення стабільної адресації.

Посилання

Це джерела виділених збігів у вашому документі. Кожен збіг позначено темно-зеленим числом, яке відповідає вказаному тут джерелу. Джерела впорядковані за схожістю — чим вищий бал, тим сильніше збіг.

#	Джерело	%
1	194.44.152.155	11.2%
2	lib.pu.if.ua	8.8%
3	194.44.152.155	1.1%
4	tpgnpu.ho.ua	0.3%
5	ts2.space	0.2%
6	julienflorkin.com	0.2%
7	conf.nikolaeviuscivile.org.ua	0.1%
8	kolo.news	0.1%
9	itgip.org	0.1%
10	alexhost.com	0.1%
11	jeldorprofi.ru	0.0%

Plag

Дякуємо, що перевірили
свій документ за допомогою
Plag!