

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету)
Кафедра прикладної математики і інформатики
(повна назва кафедри)

«До захисту допущено»

Завідувачка кафедри ПМІ



(підпис)

Ольга ДМИТРИЄВА

(ім'я та прізвище)

Випускна кваліфікаційна робота
бакалавра
(освітній ступінь)

на тему «Система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств».

зі спецчастиною «Розробка системи підрахунку кількості людей на основі датчиків платформи ARDUINO»

Виконав: студент 4 курсу, групи КІБ-18

(шифр групи)

спеціальності (напряму підготовки) 125 «Кібербезпека»

(шифр і назва спеціальності)

Білобородов Є. О.

(прізвище та ініціали)



(підпис)

Керівник проф. каф. ПМІ, д.т.н., проф Євген БАШКОВ

(посада, науковий ступінь, вчене звання, ім'я та прізвище)



(підпис)

Рецензент доц. каф. КІ, к.т.н., Юлія ДІКОВА

(посада, науковий ступінь, вчене звання, ім'я та прізвище)



(підпис)

Нормоконтроль:

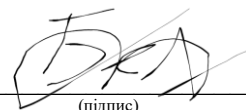


(підпис)

к.т.н., доц. Ірина НАЗАРОВА

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент



(підпис)

Луцьк – 2022 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики
Освітньо-кваліфікаційний рівень (освітній ступінь) бакалавр
Напрямок підготовки (спеціальність) 125 Кібербезпека
(шифр і назва)

ЗАТВЕРДЖУЮ:
Завідувач кафедри ПМ та І

О.А. Дмитрієва
“ ” 2022 року

З А В Д А Н Н Я **НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Білобородову Євгену Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи »Система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств»

Спецчастина »Розробка системи підрахунку кількості людей на основі датчиків платформи ARDUINO»

Керівник роботи Башков Є.О., д.т.н., проф.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від «06» травня 2022 року № 180

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____

1) результати переддипломної практики, система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств будинку на основі пристроїв ІЧ випромінювання;

2) платформа Arduino для розробки системи для розробки системи контролю кількості присутніх в приміщенні офісі;

3) розробка системи підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств будинку на основі датчиків платформи ARDUINO;

4) програмне забезпечення системи підрахунку людей в кімнатах офісу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): _____

1) огляд проблеми по застосуванню системи підрахунку кількості люд приміщеннях;

2) технічне обладнання для розробки системи підрахунку людей в приміщеннях підприємства;

3) розробка мережі системи підрахунку людей в приміщеннях офісу.

Графічний матеріал непотрібен

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Вступ		
2	Огляд проблеми по застосуванню системи підрахунку кількості людей в приміщеннях		
3	Технічне обладнання для розробки системи підрахунку людей в приміщеннях підприємства		
4	Розробка мережі системи підрахунку людей в приміщеннях офісу		
5	Висновки		
6	Список використаних джерел		

Студент


(підпис)

Білобородов Є.О.
(прізвище та ініціали)

Керівник роботи


(підпис)

Башков Є.О.
(прізвище та ініціали)

АНОТАЦІЯ

Білобородов Є. О. Система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств / Випускна кваліфікаційна робота на здобуття освітньо-кваліфікаційного рівня «бакалавр» за напрямом підготовки 125 «Кібербезпека». – ДВНЗ ДонНТУ, Луцьк, 2022.

Текстова частина бакалаврської роботи: 78 с., 22 рис., 30 джерел.

Об'єкт дослідження – процеси підтримки інформаційної безпеки малих підприємств.

Предмет дослідження – системи підтримки інформаційної безпеки малих підприємств на основі контролю кількості людей в приміщеннях.

Мета роботи полягає в розробці системи підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств.

Методи розробки базуються на основі положень теорії передавання даних та електромагнітного поля, мікропроцесорної техніки, схемотехніки та програмування на мові C++.

В бакалаврській роботі розроблено мережну систему підрахунку кількості людей в кімнатах офісу. Проведено аналіз основних моделей побудови програмно-апаратної системи та мережі на основі інфрачервоних датчиків. Обґрунтована вибір елементної бази та розроблено принципіальну електричну схему мережі лічильників на основі платформи Arduino.

Новизна роботи полягає у розвитку перспективних підходів підтримки інформаційної безпеки малих підприємств на основі удосконалення системи підрахунку кількості людей в приміщеннях.

Практичне значення полягає в розробці системи підрахунку кількості людей в приміщеннях підприємств.

Ключові слова: безпека, система підрахунку, інформаційна безпека, платформа ARDUINO, програма, датчики руху.

ЗМІСТ

	Стор.
Перелік умовних позначень, символів, скорочень і термінів	7
Вступ	8
1 Огляд проблеми по застосуванню системи підрахунку кількості людей в приміщеннях	9
1.1 Аналіз доцільності застосування системи підрахунку людей в приміщеннях	11
1.2 Аналіз існуючих технологій системи підрахунку кількості людей в приміщеннях	11
1.3 Системи підрахунку кількості людей в приміщеннях відомих виробників	17
1.3 Висновок за розділом	21
2 Технічне обладнання для розробки системи підрахунку людей в приміщеннях підприємствах	22
2.1 Стандарту RS-485 для створення мережі системи підрахунку людей в приміщеннях	22
2.2 Обладнання для реалізації мережі на основі RS485.....	25
2.3 Мікроконтролер ARDUINO Leonardo для управління мережею.....	27
2.4 Лічильник підрахунку людей на основі інфрачервоного далекоміра GP2Y0A02YK	32
2.5 Мікроконтролер ARDUINO UNO для управління лічильниками	34
2.6 Висновок за розділом.....	37
3 Розробка мережі системи підрахунку людей в приміщеннях офісу	38
3.1 Основні відомості про план розташування приміщень офісу	38
3.2 Вибір типу лічильників та способу підключення.....	39
3.3 Комплектування та монтаж обладнання підрахунку людей в приміщеннях офісу.....	42
3.4 Алгоритм роботи мережі та розробка програмного забезпечення.....	45

3.5 Блок схеми програмного забезпечення системи підрахунку кількості людей в приміщеннях	52
3.6 Висновок за розділом	59
Висновки	60
Перелік посилань.....	61
Додаток А Зауваження нормоконтролера.....	64
Додаток Б Програма для мікропроцесорних плат ARDUINO Leonardo та ARDUINO UNO	65
Додаток В Перелік копій демонстраційного матеріалу.....	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
СКОРОЧЕНЬ І ТЕРМІНІВ

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

ISO – міжнародна організація по стандартизації

PML – Physical Markup Language

XML – Extensible Markup Language

ONS – Object Name Service

СККД – система контролю та керування доступом

R/W – Read / Write

WORM – Write Once Read Many

R/O – Read Only

ВСТУП

В даний час забезпечення інформаційної безпеки малих підприємств може бути досягнути за рахунок впровадження системи контролю кількості людей в приміщеннях. Це система реалізується за допомогою арифметичного підраховування кількості людей, що заходять та виходять в приміщення. Система дозволяє контролювати загальну кількість людей, що знаходяться в офісі, а також окремо в кожній кімнаті. Перевагою також система є можливість виявляти несанкціоноване знаходження осіб в кімнатах офісу або підприємства.

Мета роботи полягає в розробці система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств.

З урахування мети роботи потрібно виконати наступні завдання:

- 1) проведено аналіз основних моделей побудови програмно-апаратної системи та мережі контролю доступу на основі інфрачервоних датчиків;
- 2) розроблено система контролю кількості присутніх в кожній кімнаті офісу;
- 3) обґрунтовано доцільність застосовування платформи Arduino.

Методи розробки базуються на основі положень теорії передавання даних та електромагнітного поля, мікропроцесорної техніки, схемотехніки та програмування на мові C++.

Обґрунтованість розробленого пристрою забезпечується відповідними схемами та програмами, що демонструють його працездатність.

В бакалаврській роботі розроблена система підрахунку кількості людей в приміщеннях для підтримки інформаційної безпеки малих підприємств на основі платформи ARDUINO.

1 ОГЛЯД ПРОБЛЕМИ ПО ЗАСТОСУВАННЮ СИСТЕМИ ПІДРАХУНКУ КІЛЬКОСТІ ЛЮДЕЙ В ПРИМІЩЕННЯХ

1.1 Аналіз доцільності застосування системи підрахунку людей в приміщеннях

Один із шляхів вирішення проблеми інформаційної безпеки малих підприємств є застосування системи підрахунку кількості людей в приміщеннях. Така система призначена для постійного контролю кількості людей, які пройшли через певний прохід за певний проміжок часу. Важливим також є визначення трафіку напрямку руху людей в різні приміщення. Слід відзначити, що найчастіше в системах підрахунку кількості людей обмежуються розподілом тих, що переміщуються, на два класи: ті, що входять, та, ті що виходять. Особливою проблемою таких систем є забезпечення точності підрахунку людей в приміщеннях. Точність підрахунку безпосередньо залежить від досконалості використовуваної технології. Система підрахунку, як правило, встановлюється на вході до приміщення, дозволяючи стежити за кількістю людей. У пристроях для підрахунку відвідувачів використовують різні технології, такі як інфрачервоні промені, комп'ютерний зір, теплобачення.

Необхідність підрахунку людей полягає в наступному (рис. 1.1). Існують різні причини, через які організації може знадобитися постійний підрахунок людей.

Одним із важливих факторів інформаційної безпеки підприємства є контроль за ефективністю роботи персоналу. Оцінити якість та ефективність роботи персоналу допомагає аналітична складова системи підрахунку людей в приміщеннях підприємствах. Якщо кількість відвідувачів набагато більше кількості покупців, необхідно переглянути стратегії продажу чи обслуговування, вводячи різні нововведення у продажу та вказуючи їх у системі обробки даних для подальшого аналізу та виявлення причин.



Рисунок 1.1 – Необхідність встановлення системи підрахунку людей

Джерело: авторська розробка

Аналіз ефективності торгівлі є складовою забезпечення як економічної безпеки, так і інформаційної. Підрахунок відвідувачів на торгових майданчиках необхідний обчислення частки покупців серед загальної кількості відвідувачів магазину. Це дуже важливий показник, який реально демонструє ефективність роботи магазину. Аналізуючи продажі важливо зрозуміти, яка кількість людей склала ці продажі, ще важливіше стежити за часткою відвідувачів, які не зробили жодної покупки. Якщо все більший відсоток людей йдуть без покупок, необхідно терміново шукати причини та усувати їх.

Таким чином, можна вважати, що лічильники відвідувачів є зручним маркетинговим інструментом, який дозволяє власнику підприємства враховувати кількість відвідувань торговельної площадки.

Переглядаючи графіки статистики відвідувань, можна виявляти та виключати різні фактори, що впливають на відвідуваність, наприклад, маркетингові акції або обставини, що знижують відвідування, такі як низький рівень обслуговування, поганий асортимент, брак реклами.

Застосування системи контролю підрахунку людей в приміщення дозволяє більше ефективно складати розклад роботи персоналу: охоронники, торговельні працівники, адміністратори та інші. Таким чином, на торгових площадках, адміністративних комплексів та інших місцях масового обслуговування потреба у персоналі може бути прямо пропорційна кількості

відвідувачів. Тому точний підрахунок відвідувачів необхідний для складання оптимального розкладу роботи співробітників.

Розглянемо систему підрахунку кількості людей в приміщеннях з точку зору планування службових заходів. Такі заходи як прибирання приміщення або інші технічні роботи повинні проводитися під час мінімального потоку відвідувачів. Лічильник відвідувачів надає необхідну статистичну інформацію, яка в даному випадку використовується для того, щоб мінімізувати незручності самої організації та її клієнтів.

Розглянемо систему підрахунку кількості людей в приміщеннях з точку зору забезпечення безпека. Для багатьох організацій важливо знати точну кількість людей, які зараз перебувають усередині будівлі, щоб при необхідності їх оперативної евакуації. Лічильник відвідувачів дозволить оцінити необхідний обсяг транспорту та інших засобів, але визначити зі 100% точністю чи всі люди покинули приміщення жоден лічильник не в змозі через похибку.

З фінансової точки зору система контролю підрахунку кількості людей в приміщення потрібно для аналізу використання ресурсів. Багато громадських організацій здійснюють підрахунок відвідувачів перед тим, як зробити запит на фінансування. У випадках, коли квитки не продаються (як у бібліотеках та музеях), необхідно автоматизувати підрахунок відвідувачів.

1.2 Аналіз існуючих технологій системи підрахунку кількості людей в приміщеннях

Сучасні системи підрахунку відвідувачів створюються на основі різних технологій (рис. 1.2), наприклад, інфрачервоні промені або ультразвук. Кожна з цих технологій має свої переваги та недоліки.

Лічильники, що працюють на інфрачервоних променях, бувають декількох типів – вертикальні та горизонтальні.

Горизонтальні лічильники працюють за принципом переривання інфрачервоного проміння. У цих лічильниках обов'язковими пристроями є

приймач та передавач. Передавач випромінює інфрачервоний промінь, а приймач приймає. Після перетину промінню приймач аналізує факт проходження. Існують наступні прилади підрахунку кількості людей у приміщенні. Це ті, які як правило підключаються до персонального комп'ютера (ПК), або прилад працює в автономному режимі. У першому варіанті дані виводяться на ПК, у другому на РК дисплей лічильника.

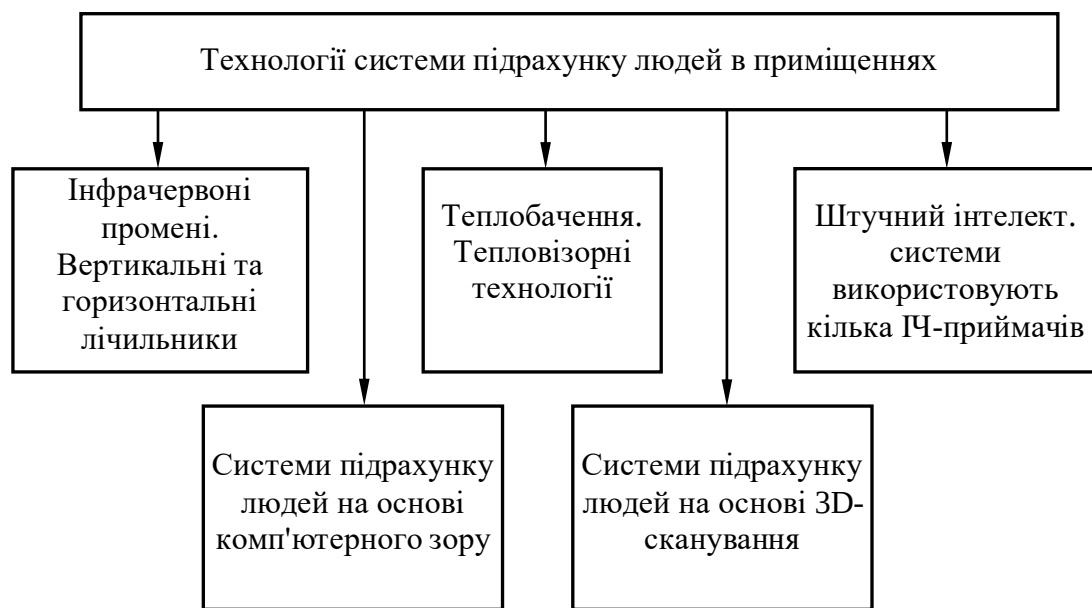


Рисунок 1.2 –Технології системи підрахунку людей в приміщеннях

Джерело: авторська розробка

Лічильники, що відображають дані на дисплеї мають ряд властивостей, Вони можуть працювати за рахунок автономного живлення. Також лічильники можуть мати зв'язок з центром охорони по радіоканалу або проводу, тобто це означає фізичний зв'язок з ПК. Також можливо застосування для зв'язку з лічильником технологій WiFi або GSM. Лічильники можуть бути двонаправленими або односпрямованими, тобто є різновиди напрямку руху даних, що передається. Таким чином, при розробці лічильника кількості підрахунку людей потрібно визначити основні вимоги до нього.

Лічильники покупців, які підключаються до ПК, піддаються простий автоматизації збору даних для аналізу. У простому варіанті дані

проглядаються на локальному ПК. Коли є мережа торгових точок, дані передаються через інтернет до центрального офісу з усіх точок. Як правило, сервер це безпечний ресурс на потужностях майкрософт, де можна переглядати дані з будь-якого ПК у веб-браузері. Рідше на локальному за бажанням клієнта [5].

Переваги лічильників на основі інфрачервоних променях наступні:

- недорогі;
- прості в установці;
- точність у проході до 1.5м – 98% (100 люд/год) ;
- фіксує факт перекриття променю та вимикання живлення;
- підраховують точно при вході.

Недоліки лічильників на основі інфрачервоних променях полягаю в тому, що похибка зростає при збільшенні потоку та зростанні ширини проходу, оскільки зростає можливість проходу на одній лінії двох відвідувачів. Похибка підрахунку окремих випадках може досягати 10 % [2].

Вертикальні лічильники працюють за принципом відображення променю тіла людини. Теоретично вертикальні промені точніші, ніж горизонтальні, і разі застосування сучасних моделей можна досягти високої точності підрахунку людей, до 95 % і від. Однак, це справедливо не для всіх модифікацій. Більшість вертикальних інфрачервоних сенсорів є «рампу» встановлену над проходом, в якій на відстані близько 50-60 см розташовуються вертикальні інфрачервоні сенсори.

Алгоритм простий – перетин одного променю – одному відвідувачеві. У зв'язку з цим такі системи можуть давати дуже високі похибки, так як ширина плечей людини коливається в діапазоні 35-70 см і більше (взимку).

Як правило, більшість горизонтальних і вертикальних інфрачервоних сенсорів дозволяє вести статистику відвідуваності на ПК користувача.

Основні недоліки вертикальних лічильників:

- висока ціна;
- низька точність.

Розглянемо лічильники на основі тепlobачення. Системи тепlobачення використовують спеціальні сенсори. Подібні системи зазвичай реалізуються як убудовані пристрої. Є одним із найбільш точних сенсорів. Висока точність підрахунку обумовлена використанням тепловізорної технології, яка визначає людину за її тепловою плямою та аналізує рух у зоні детекції.

Переваги системи тепlobачення:

- висока точність;
- простота встановлення;
- інформація про напрямок – не залежить від умов освітленості та здатна вести підрахунок у повній темряві.

Недоліки системи тепlobачення:

- вища ціна, ніж у «променевих» систем;
- більш складна реалізація, ніж у «променевих» систем;
- значне падіння точності в умовах температурного фону, що змінюється (теплові гармати, пересування повітряних мас).

Системи підрахунку людей на основі штучний інтелекту - подібні системи використовують кілька ІЧ-приймачів для того, щоб створити «зону підрахунку» на рівні кісточки. Інтелектуальні лічильники функціонують подібно до людського мозку, іншими словами, кожна подія аналізується в термінах «ознак». Коли «зона підрахунку» перетинається кимось, генерується патерн (шаблон). Вбудований процесор отримує ознаки даного шаблону і робить його розпізнавання: скільки і в яких напрямках були проходи.

Переваги системи підрахунку людей на основі штучного інтелекту:

- точність 90% і вище;
- інформація про напрямок руху;
- здатність відрізнати людей від інших об'єктів;
- сенсори здатні вести підрахунок поза приміщенням;
- здатність вести підрахунок у будь-яких умовах освітленості.

Недоліки системи підрахунку людей на основі штучного інтелекту:

- висока ймовірність бути заблокованими людьми чи предметами;

- система має великі габарити;
- на підрахунок впливає перепад освітлення, відблиски;
- при низькій освітленості точність значно знижується;
- неможливість робити підрахунок у розважальних спеціально затемнених зонах.

Системи підрахунку людей на основі комп'ютерного зору є одна з найпопулярніших технологій, яка має широке застосування на ринку. Технології комп'ютерного зору дозволяють створити лічильник відвідувачів найвищої якості. Лічильник розпізнає проходи людей у відеопотоці, що надходить із двох web-камер, об'єднаних в одному корпусі пристрою та встановлених над входом до приміщення. Завдяки двом камерам створюється ефект стереовідеозображення, що позитивно позначається на точності підрахунку порівняно з сенсорами, що використовують лише одну web-камеру. Деякі системи такого класу інтегруються з базами даних про продаж, що дозволяє здійснювати повноцінний аналіз.

Точність підрахунку відвідувачів може коливатися від приміщення до приміщення. Системи, що базуються на комп'ютерному зорі, чутливі до зміни освітлення та тіней, які можуть призвести до неточного підрахунку. Якщо освітленість приміщення постійна, як правило досягається висока точність підрахунку, однак, наприклад, підрахунок людей на вулиці може бути неточним через потрапляння сонячного світла в камеру або навпаки, падіння освітленості через хмари.

Переваги системи підрахунку людей на основі комп'ютерного зору:

- висока точність, більше 95 %;
- дані про зростання;
- гнучкість при налаштуванні під конкретне приміщення;
- інтеграція з іншими системами;
- вбудована система відеоспостереження.

Недоліки системи підрахунку людей на основі комп'ютерного зору:

- точність залежить від коливань рівня освітленості;

- на точність можуть впливати такі фактори як відблиски, відбиття від глянсової статі;

- при низькій освітленості точність значно знижується;

- неможливість робити підрахунок у розважальних спеціально затемнених зонах.

В системах підрахунку людей на основі 3D-сканування використовуються 3D-сенсори, на базі кінект. Вони встановлюються на стелі приміщення, іноді під невеликим кутом, і проектує на сітку інфрачервоних променів, що спостерігається, з більше мільйона точок. Таким чином зчитується рельєф області, що спостерігається, після чого процесором сенсора будується повноцінна 3D-модель спостерігається області. Це позбавляє сенсор від появи похибки внаслідок збігу кольору одягу відвідувачів, що трапляється із сенсорами відеотехнологій та технологій комп'ютерного зору. Не варто недооцінювати цю властивість 3D-сенсора: в осінньо-зимовий період переважають темні кольори, що може призвести до суттєвого зростання похибки у сенсорів інших технологій.

Застосовані технології дозволяють використовувати сенсор у різних умовах: при різній освітленості, включаючи повну відсутність світла (що дозволяє застосовувати його навіть у кінотеатрах і розважальних центрах), за наявності повітряних потоків і таке інше.

Переваги системи підрахунку людей на основі 3D-сканування:

- висока точність, приблизно 99%;

- широкі можливості застосування у приміщеннях різної архітектурної складності;

- незалежність від зовнішніх умов експлуатації, таких як рівень освітленості та тепловий фон приміщення;

- підходить для підрахунку навіть дуже щільних потоків людей, у тому числі і для встановлення на головних входах до торгового центру;

- колір не впливає на підрахунок;

- віддалене діагностування та настройка обладнання.

Недоліки системи підрахунку людей на основі 3D-сканування;

- досить висока вартість;
- практично перестає працювати при яскравому сонячному освітленні або яскравому ламповому підсвічуванні;
- неможливість встановлення на вулиці;
- 3D-ASSIS.

Вітчизняна розробка технології підрахунку відвідувачів полягає у обробці двовимірного відео потоку з перетворенням їх у тривимірну математичну модель представлення.

1.3 Системи підрахунку кількості людей в приміщеннях відомих виробників

Лідером європейського ринку в реєстрації статистики систем підрахунку відвідувачів є британська компанія FootFall – її лічильники щорічно реєструють близько 6 мільярдів осіб. Найбільший відомий виробник комплексної системи підрахунку відвідувачів – MegaCount [7]. Станом на 2015 найбільшими інтеграторами на ринку є компанії MegaCount, Перший Біт, Watcom group, АНТИвор [8,9]. Для аналізу відвідувальних потоків у магазинах та торгових центрах використовують лічильники відвідувачів. Вони призначені для розрахунку показників конверсії шляхом порівняння кількості відвідувачів з кількістю покупців. Дають можливість об'єктивно оцінити навантаження на персонал, ефективність реклами, піковий годинник, лінії тренду та популярні області. На ринку лічильники підрахунку представлені в широкому асортименті, проте всі вони мають свої переваги і недоліки.

За принципом роботи розрізняють три основні групи: інфрачервоні лічильники; теплові датчики; відеоаналітичні системи.

Системи, що здійснюють підрахунок людей за допомогою інфрачервоних променів (рис. 1.3, 1.4). Такі системи складаються з джерела та приймача ІЧ променів, що розриваються людиною, під час його руху.

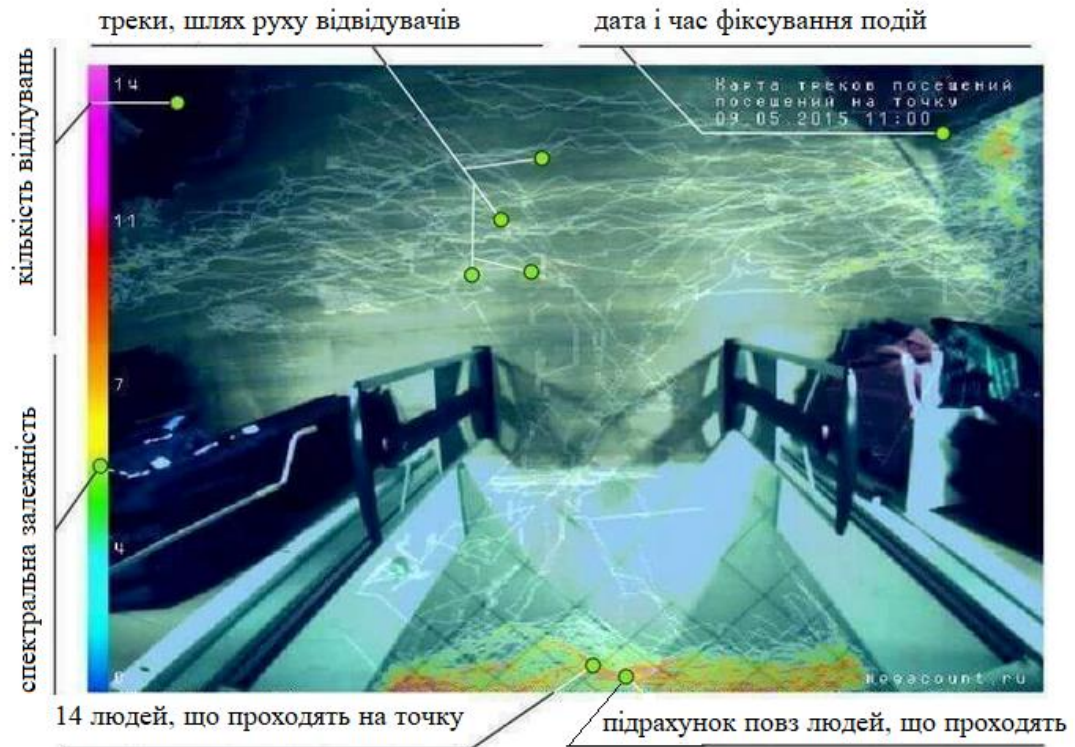


Рисунок 1.3 – Принцип дії інфрачервоних лічильників людей в приміщеннях

Джерело [22]

Горизонтальні датчики можуть бути однолінійні та дволінійними. Однолінійні лічильники нездатні розрізнати напрямок руху, оскільки мають один промінь. Дволінійні лічильники мають два промені. Завдяки різній послідовності їх перетину, вони підраховують кількість входів та виходів. Залежно від способу подачі живлення, такі датчики бувають автономні, що працюють від батарейок, і неавтономні, що підключаються до електромережі.

Також існують моделі, що підтримують і не підтримують підключення до комп'ютера.



Рисунок 1.4 – Пристрій інфрачервоного лічильника людей

Джерело [23]

За способом монтажу розрізняють горизонтальні, з допустимою шириною проходу, згідно з технічною документацією, менше 5 м і вертикальні, з висотою установки до 3 м. Можна відзначити наступні їх переваги: низька вартість; легкий монтаж та налаштування; оперативність аналітичних звітів та передавання даних через мережу. Проте до недоліків можна віднести наступне: висока похибка під час підрахунку людей; не підходять для установки в широкий прохід; без підключення до комп'ютера, неможливо переглядати часові графіки, доступні лише денні звіти; легко обійти систему підрахунку; збій програмного забезпечення, яке складно перевстановити, не звертаючись до постачальника.

В цілому інфрачервоні лічильники недорогі, але не підходять для широкого проходу і дають показання з похибкою, яка залежить від ширини проходу, до того ж багато хто з них примхливі у використанні з комп'ютером. Проте вони добре підходять для отримання приблизних показників, за малої протоки відвідувачів.

Горизонтальні ІЧ-лічильники підрахунку людей складаються з двох датчиків – передавача та приймача, які кріпляться у проходах на рівні 1,2-1,4

метра від підлоги. При цьому фотоелементи використовуються в якості приймальних датчиків. Працюють такі системи за принципом переривання інфрачервоного променя: передавач випромінює безперервний ІЧ-сигнал, який приймається приймачем.

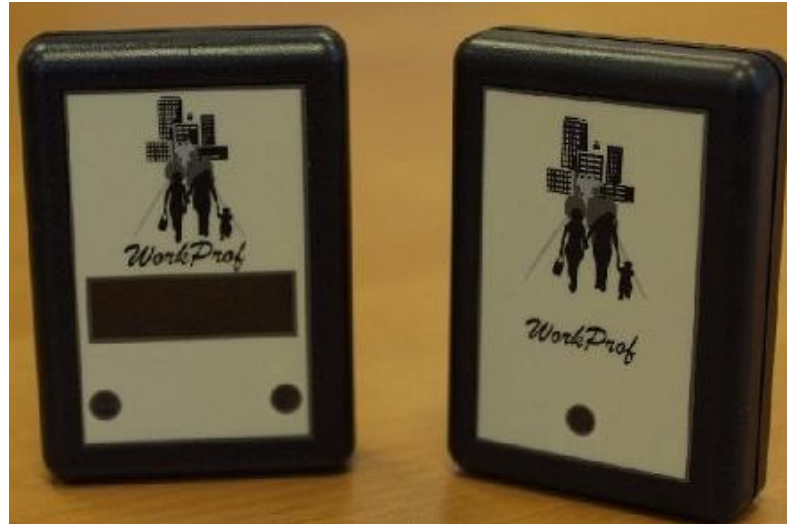


Рисунок 1.5 – Горизонтальні ІЧ-датчик системи підрахунку відвідувачів
Джерело [24]

Частково вирішити проблему зниження точності можна використанням вертикальних ІЧ-лічильників, проте це справедливо далеко не для всіх модифікацій. На відміну від вертикальних, горизонтальні системи підрахунку покупців розташовуються над проходом і працюють за принципом відображення інфрачервоного променя тіла людини (при перетині одного променя фіксується прохід одного відвідувача).

Один з найбільш поширених варіантів горизонтальних ІЧ-лічильників є рампа, в якій ІЧ-сенсори розташовуються на відстані 50-60 см один від одного. Точність систем такого типу залежить від ширини проходу. Зате від неї залежить ціна лічильника: що ширший прохід, то більше в ньому потрібно розмістити інфрачервоних сенсорів і тим дорожче вийде кінцева вартість системи. Крім того, в порожньому проході в залежності від ширини плечей і

траєкторії руху одна людина може перетнути два, три промені або пройти між променями, тому похибка підрахунку в окремих випадках може досягати 20%.

Таким чином, з урахуванням усіх зазначених переваг і недоліків вертикальні ПЧ-лічильники найкраще підходять для офісних кімнат та торгових об'єктів з вузькими проходами і невеликою інтенсивністю руху, а горизонтальні дозволяють вирішити проблему підрахунку потоків відвідувачів великої інтенсивності в широких проходах.

1.4 Висновок за розділом

Результати аналізу першого розділу дали змогу зробити висновки, які пов'язані з розробкою системи підрахунку кількості людей в приміщення з метою організації інформаційної безпеки підприємства, а також інших складових, що пов'язані з маркетинговими задачами.

Обґрунтована доцільність застосування системи підрахунку людей в приміщеннях. Розглянуто технології систем підрахунку кількості людей в приміщеннях. Надано аналіз систем підрахунку кількості людей в приміщеннях відомих виробників.

Визначено, що вертикальні ПЧ-лічильники кількості людей найкраще підходять для офісних кімнат та торгових об'єктів з вузькими проходами і невеликою інтенсивністю руху, а горизонтальні дозволяють вирішити проблему підрахунку потоків відвідувачів великої інтенсивності в широких проходах.

2 ТЕХНІЧНЕ ОБЛАДНАННЯ ДЛЯ РОЗРОБКИ СИСТЕМИ ПІДРАХУНКУ ЛЮДЕЙ В ПРИМІЩЕННЯХ ПІДПРИЄМСТВАХ

2.1 Стандарт RS-485 для створення мережі системи підрахунку людей в приміщеннях [2, 21]

RS-485 (англ. Recommended Standard 485), EIA-485 (англ. Electronic Industries Alliance-485) [2, 21] є стандарт фізичного рівня для асинхронного інтерфейсу. Назва стандарту: ANSI TIA/EIA-485-A:1998 Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems. Він регламентує електричні параметри напівдуплексної багатоточкової диференціальної лінії зв'язку типу "загальна шина". Також цей стандарт є основою для створення цілого сімейства промислових мереж, які широко використовуються в промисловій автоматизації.

Стандарт RS-485 спільно розроблений двома асоціаціями: Асоціацією електронної промисловості (EIA – Electronic Industries Association) та Асоціацією промисловості засобів зв'язку (TIA – Telecommunications Industry Association). Раніше EIA маркувала всі свої стандарти префіксом RS (англ. Recommended Standard – Рекомендований стандарт). Багато інженерів продовжують використовувати це позначення, проте EIA/TIA офіційно замінив RS на EIA/TIA з метою полегшити ідентифікацію походження своїх стандартів.

Розглянемо технічні характеристики RS-485. У стандарті RS-485 для передачі та прийому даних використовується одна кручена пара проводів, іноді супроводжується екрануючим обплетенням або загальним проводом.

Передача даних здійснюється з допомогою диференціальних сигналів. Різниця напруги між провідниками однієї полярності означає логічну одиницю, різниця іншої полярності - нуль.

Стандарт RS-485 обумовлює лише електричні та часові характеристики інтерфейсу. Стандарт RS-485 не застерігає:

- параметри якості сигналу (допустимий рівень спотворень, відображення у довгих лініях);
- типи з'єднувачів та кабелів;
- гальванічну розв'язку лінії зв'язку;
- протокол обміну.

Електричні та часові характеристики інтерфейсу RS-485:

- підтримуються до 32 приймачів в одному сегменті мережі;
- максимальна довжина одного сегмента: 1200 метрів;
- одночасно активним може бути лише один передавач;
- максимальна кількість вузлів у мережі – 256 з урахуванням магістральних підсилювачів.

Співвідношення швидкість обміну/довжина лінії зв'язку [1]:

- 62,5 кбіт/с 1200 м (одна кручена пара);
- 375 кбіт/с 500 м (одна кручена пара);
- 500 кбіт/с;
- 1000 кбіт/с;
- 2400 кбіт/с 100 м (дві кручені пари);
- 10 000 кбіт/с 10 м-коду.

Тип приймачів – диференціальний, потенційний. Зміна вхідної та вихідної напруги на лініях А і В: U_a (U_b) від -7 В до -12 В ($+7$ В до $+12$ В).

Вимоги до вихідного каскаду:

- вихідний каскад є джерелом напруги з малим вихідним опором, $|U_{вых}|=1,5;5,0$ В (не менше 1,5 В і не більше 5,0 В);
- стан логічного «1»: $U_a < U_b$ (гістерезис 200 мВ) - MARK, OFF;
- стан логічного «0»: $U_a > U_b$ (гістерезис 200 мВ) – SPACE, ON (виробники мікросхем – драйверів, часто вибирають набагато менші значення, гістерезис від 10 мВ [2, 21]);
- вихідний каскад повинен витримувати режим короткого замикання, мати максимальний вихідний струм 250 мА, швидкість наростання вихідного сигналу 1,2 В/мкс та схему обмеження вихідної потужності.

Вимоги до вхідного каскаду:

- вхідний каскад є диференціальним входом з високим вхідним опором і пороговою характеристикою від -200 мВ до $+200$ мВ;
- допустимий діапазон вхідної напруги U_{ag} (U_{bg}) щодо землі (GND) від -7 до $+12$ В;
- вхідний сигнал представлений диференціальною напругою ($U_i +0,2$ і більше);
- рівні стану приймача вхідного каскаду — див. стан передавача вихідного каскаду.

При довжині лінії зв'язку виникають ефекти довгих ліній. Причина цього - розподілені індуктивні та ємнісні властивості кабелю. Як наслідок, сигнал, переданий до лінії одним з вузлів, починає спотворюватися в міру поширення в лінії, виникають складні резонансні явища. Оскільки на практиці кабель на всій довжині має однакову конструкцію і, отже, однакові розподілені параметри погонної ємності та індуктивності, цю властивість кабелю характеризують спеціальним параметром – хвильовим опором. Не вдаючись у теоретичні подробиці, можна сказати, що в кабелі, на приймальному кінці якого підключено узгоджене навантаження (резистор з опором, що дорівнює хвильовому опору кабелю), резонансні явища значно послаблюються. Називається такий резистор термінатором. Для мереж RS-485 вони ставляться на кожному кінці довгої лінії (оскільки обидві сторони можуть бути приймальними). Хвильовий опір найпоширеніших кручених пар типу CAT5 становить 100 Ом [2, 21]. Інші кручені пари можуть мати хвильовий опір 150 Ом і вище. Плоскі стрічкові кабелі до 300 Ом [2, 21].

На практиці номінал цього резистора може вибиратися і більшого номіналу, ніж хвильовий опір кабелю, оскільки омичний опір того ж кабелю може виявитися настільки велике, що амплітуда сигналу на приймальній стороні виявиться занадто мала для сталого прийому. У цьому випадку шукають компроміс між резонансними та амплітудними спотвореннями сигналу, зменшуючи швидкість інтерфейсу та збільшуючи номінал

термінатора [2, 21]. На швидкостях 9600 біт/с і нижче хвильові, резонансні явища у масштабах, здатних погіршити якість зв'язку, не виявляються, і питання узгодження лінії не виникає. Навіть більше, при низьких швидкостях передачі (менше 9600 біт/с) термінальний резистор не покращує, а погіршує надійність передачі (істотно для довгих ліній зв'язку) [2, 21].

Ще одне джерело перекручування форми сигналів під час передачі через кручену пару - різна швидкість поширення високочастотного і низькочастотного сигналу (високочастотна складова поширюється по крученій парі дещо швидше), що зумовлює спотворення форми сигналу при високих швидкостях передавань.

Перешкоди в лінії зв'язку залежать не тільки від довжини, термінаторів та якості крученої пари. Важливо, щоб лінія зв'язку послідовно оминала всі приймачі (топология загальної шини). Віта пара не повинна мати довгих відводів - відрізків кабелю для з'єднання з черговим вузлом, крім випадку використання повторювачів інтерфейсу, або при низьких швидкостях передачі менше 9600 біт/с.

У момент відсутності активного передавача на шині, рівень сигналу в лініях не визначений. Для запобігання ситуації, коли різниця між входами А і В менша за 200мВ (невизначений стан), іноді застосовується зміщення за допомогою резисторів або спеціальної схеми. Якщо стан ліній не визначено, приймачі можуть приймати сигнал завади. Деякі протоколи передбачають передачу службових послідовностей для стабілізації приймачів та впевненого початку прийому.

2.2 Обладнання для реалізації мережі на основі RS485

Для розробки мережі офісу для підрахунку кількості відвідувачів в різних приміщеннях можна застосувати плату RS-485 Shield (рис. 2.1). За допомогою цієї плати можна розгортати масштабні мережі інтелектуальних датчиків, що працюють завдяки мікроконтролеру Arduino UNO. Підключення плати RS-485 Shield до плати Arduino UNO дозволить організувати мережу на

основі протоколу RS-485 Shield за допомогою двох дротів. Компоновка плати RS-485 Shield наступна.

Плата побудована на мікросхемі Maxim MAX485 та виробляється компанією DFRobot. Мікросхема MAX485, на якій заснований плата розширення, має набір засобів захисту від зовнішніх впливів:

- захист від короткого замикання ліній;
- захист від перегріву та захист від електростатичних розрядів.

На платі є майданчик із отворами для прототипування. Крок отворів стандартний – 2,54 мм.

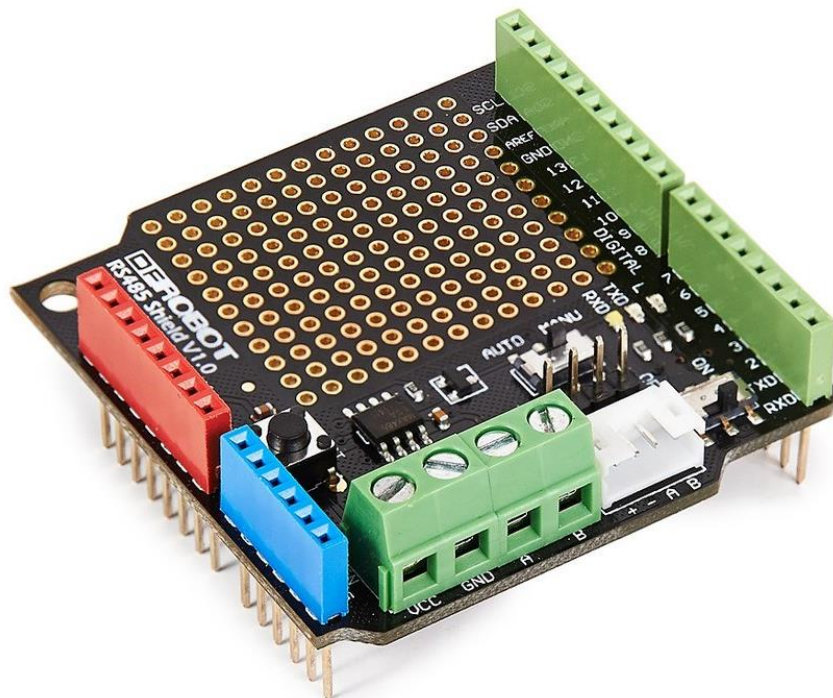


Рисунок 2.1 – Зовнішній вид плати розширення RS-485 Shield

Джерело [25]

Розглянемо особливості інтерфейсу RS485, який є симплексний. Це означає, що у будь-який час дані можуть передаватися лише одним вузлом мережі. Причому дані передаються відразу по двох проводах диференціальним методом: біт, що передається, кодується різницею потенціалів між двома проводами лінії. Такий спосіб передачі значно знижує рівень перешкод, т.к. вони наводяться однаково на обидва дроти і «не видно»

при вимірі різниці потенціалів на цих дротах. Як наслідок, пристрої можуть спілкуватися на відстані до 1200 метрів.

Передбачено два способи перемикання між передачею та прийомом: автоматичний та ручний. В автоматичному режимі плата сама визначає, коли контролер починає видавати дані та перемикається на передачу. Режим встановлюється перемикачем плати.

Також на платі встановлено вимикач обміну даними. Він просто від'єднує модуль від лінії. Ця функція виявляється дуже корисною на етапі налагодження.

Плати розширення немає опторозв'язки, проте можуть становити мережу, як із єдиним, і з безліччю джерел живлення. Дані передаються диференціальне, тому гальванічна розв'язка вузлів живлення не потрібна.

Використовувані піни плати RS-485 Shield. Управління платою здійснюється через UART. Тому зайнятими виявляються піни 0 (RX) та 1 (TX), що використовуються для прийому та передачі даних відповідно.

Крім того, в ручному режимі вибору напрямку передачі використовується пін 2. Установка логічної одиниці відповідають напряму на передачу, нуля – прийом.

Багато Arduino-плати, наприклад Arduino Uno, використовують ці піни для завантаження нового скетчу. Щоб інтерфейс RS-232 не заважав перепрошивці плати, потрібно встановити перемикач передбачений на платі в положення «OFF».

Буває необхідно одночасно використовувати інтерфейс RS-485 та зв'язок з комп'ютером. У такому разі як керуючий пристрій можна використовувати Arduino Leonardo, в якій апаратний UART та USB-інтерфейс рознесені.

2.3 Мікроконтролер Arduino Leonardo для управління мережею

Arduino Leonardo – це мікроконтролерна плата на базі чіпа ATmega32u4. Вона має 20 цифрових I/O контактів (з яких 7 можна використовувати для

видачі ШІМ, а 12 – як вхідні аналогові контакти), кварцовий генератор на 16 МГц, з'єднання типу micro USB, роз'єм для живлення, ICSP-гребінець та кнопка скидання. Плата Arduino Leonardo (рис. 2.2) оснащена всім потрібним для підтримки мікроконтролера; щоб почати, просто підключіть її до комп'ютера за допомогою USB-кабелю або запитайте від адаптера, що конвертує змінний струм на постійний, або від батареї.

Leonardo відрізняється від усіх моделей-попередниць тим, що її чіп ATmega32u4 має вбудований функціонал для USB-комунікації, що усуває потребу у другому процесорі. Це дозволяє Leonardo відображатися на комп'ютері як миша або клавіатура – до віртуального (CDC) послідовного COM порту. Це впливає на поведінку плати і в іншому, і про це можна прочитати на цій сторінці.

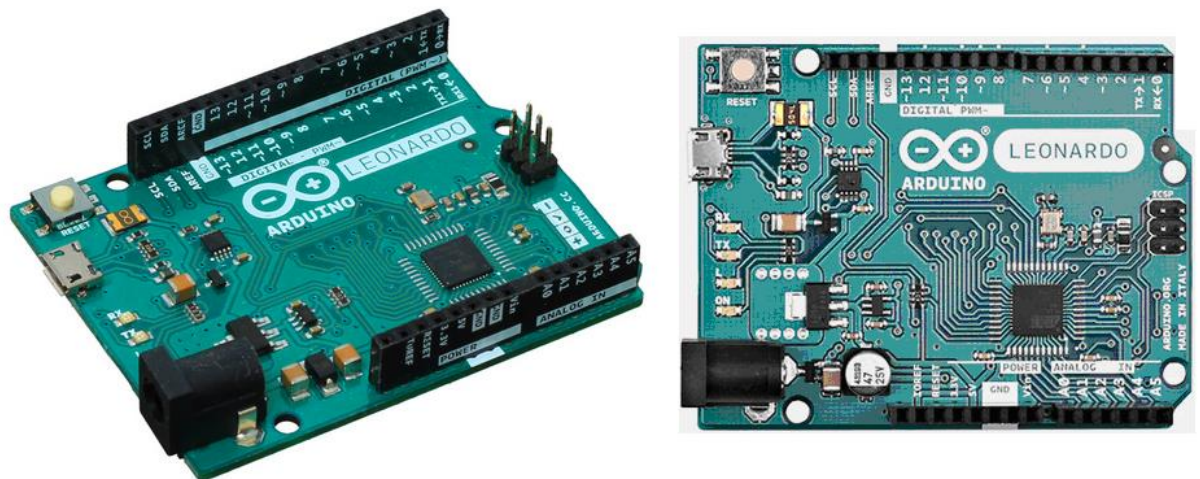


Рисунок 2.2 – Зовнішній вид плати Arduino Leonardo

Джерело [26]

Технічні характеристики плати Arduino Leonardo наступні:

- Мікроконтролер – ATmega32u4
- Робоча напруга – 5 вольт
- Вхідна напруга (рекомендована) – 7-12 вольт
- Вхідна напруга (ліміт) – 6-20 вольт

- Цифрові I/O контакти – 20 шт.
- Цифрові I/O контакти із підтримкою ШІМ – 7 шт.
- Вхідні аналогові канали – 12 шт.
- Максимальна сила струму на один I/O контакт – 40 міліампер
- Максимальна сила струму на контакт 3.3V – 50 міліампер
- Flash-пам'ять – 32 Кб (ATmega32u4), з яких 4 Кб використовуються для

завантажувача

- SRAM – 2,5 Кб (ATmega32u4)
- EEPROM – 1 Кб (ATmega32u4)
- Тактова частота – 16 МГц
- Довжина – 68,6 мм.
- Ширина – 53,3 мм.
- Вага – 20 грам

харчування

Плату Arduino Leonardo можна живити через USB-з'єднання або від зовнішнього джерела живлення. Джерело живлення вибирається автоматично.

Зовнішнє (не USB) живлення може йти або від адаптера, що конвертує змінний струм на постійний, або від батареї. Адаптер підключається до плати за допомогою 2,1-міліметрового конектора з центральним позитивним контактом або від роз'єму для живлення на платі. Провід від батареї потрібно підключити до контактів Gnd та Vin на конекторі POWER.

Контакти для харчування на Arduino Leonardo:

- VIN – вхідна напруга під час використання зовнішнього джерела живлення (на відміну від 5 вольт, що йдуть від USB-з'єднання або іншого відрегульованого джерела живлення). На цей контакт можна, з одного боку, подавати живлення, а з іншого, брати його звідти, якщо напруга подається на роз'єм для живлення.

- 5V – регульоване джерело живлення, яке використовується для живлення мікроконтролера та інших компонентів плати. Воно може йти або

від контакту VIN через вбудований регулятор, або подаватися через USB або інше регульоване джерело 5-вольтової напруги.

- 3V3 – напруга 3,3 вольт, що генерується вбудованим регулятором. Максимальна сила струму, яку можна тягнути звідси – 50 міліампер.

- GND – контакти для землі.

- IOREF – напруга, на якій працюють I/O контакти плати (тобто VCC для плати). на Leonardo це 5 вольт.

Пам'ять. Чіп ATmega32u4 має 32 Кб пам'яті для зберігання скетчів (з них 4 Кб використовуються завантажувачем). Крім того, він оснащений 2,5 Кб пам'яті типу SRAM та 1 Кб пам'яті типу EEPROM.

Запис та зчитування з пам'яті EEPROM здійснюється за допомогою бібліотеки EEPROM.

Розглянемо вхідні та вихідні контакти плати. Кожен із 20 цифрових I/O контактів Leonardo можна використовувати і як вхідний, і як вихідний контакт – за допомогою функцій `pinMode()`, `digitalWrite()` і `digitalRead()`. Вони оперують на 5 вольтах. Кожен контакт може давати/отримувати не більше 40 міліампер і має вбудований підтягуючий резистор (за замовчуванням вимкнено) на 20-50 кОм. Крім того, деякі контакти мають спеціальні функції:

- Послідовна передача даних: 0-й (RX) та 1-ий (TX) контакти. Використовуються для отримання (RX) та передачі (TX) послідовних (TTL) даних за допомогою чіпа ATmega32u4. Зверніть увагу, що для використання комунікації типу USB (CDC) у Leonardo використовується клас `Serial`, а для комунікації типу TTL, яка здійснюється на 0-му та 1-му контактах, використовується клас `Serial1`.

- Інтерфейс TWI: другий (SDA) і третій (SCL) контакти. Комунікація типу TWI, що здійснюється за допомогою бібліотеки `Wire`.

- Зовнішні переривання: 3-й (переривання 0), 2-й (переривання 1), 0-ий (переривання 2), 1-ий (переривання 3) та 7-ий (переривання 4) контакти. Ці контакти можна настроїти на запуск переривання при переході в значення

LOW, зростаючому/спадному фронту імпульсу або зміні значення. Докладніше читайте у статті про функцію `attachInterrupt()`.

- ШІМ: 3-й, 5-й, 6-й, 9-й, 10-й, 11-й і 13-й контакти. Забезпечують видачу ШІМ з допомогою функції `analogWrite()`.

- Інтерфейс SPI: на гребінці ICSP. Ці контакти здійснюють комунікацію типу SPI за допомогою бібліотеки SPI. Зверніть увагу, що контакти SPI не підключені до жодного I/O контакту, як це зроблено на Uno, і доступні лише на ICSP-конекторі. Це означає, що якщо у вас є "шилд", що використовує інтерфейс SPI, але НЕ МАЄ 6-контактного ICSP-конектора, який можна було б підключити до 6-контактного ICSP-гребінця на Leonardo, то цей "шилд" працювати не буде.

- 1) Вбудований світлодіод: 13 контакт. Якщо подати цей контакт значення HIGH, світлодіод включиться, і якщо LOW, то вимкнеться.

- 2) Вхідні аналогові контакти: з A0 по A5 (там же, де у Uno), а також з A6 по A11 (на 4-му, 6-му, 8-му, 9-му, 10-му і 12-му контактах). Плата Leonardo має 12 вхідних аналогових контактів, названих з A0 до A11, і кожен з них можна також використовувати як цифрові I/O контакти. Кожен із цих контактів підтримує 10-бітове дозвіл (т.к. дозволяє оперувати 1024 різними значеннями). За замовчуванням діапазон напруги цих контактах становить від 5 вольт до «землі», проте його верхню межу можна змінити з допомогою контакту AREF і функції `analogReference()`.

На платі Leonardo є інші спеціалізовані контакти:

- AREF – еталонна напруга для аналогових вхідних контактів. Використовується разом із функцією `analogReference()`.

- Reset – скидання мікроконтролера. Щоб скинути мікроконтролер, на цей контакт необхідно подати значення LOW. Як правило, використовується, щоб додати на шилд кнопку Reset, тому що підключення шилда блокує кнопку скидання на самій платі.

Плата Leonardo має кілька засобів для комунікації із комп'ютером, іншими Arduino або іншими мікроконтролерами. По-перше, чіп ATmega32u4

підтримує послідовну передачу типу UART TTL (5-вольтову), яка доступна на 0-му та 1-му контактах. По-друге, ATmega32u4 підтримує послідовну (CDC) комунікацію через USB, завдяки чому програмне забезпечення на комп'ютері бачить його як віртуальний COM-порт. Крім того, ATmega32u4 функціонує як пристрій типу USB 2.0 з високою пропускну здатністю (full-speed), використовуючи для цього стандартні USB/COM драйвери. Windows потребує файлу *.inf.

Платформа IDE Arduino має монітор послідовного порту, що дозволяє відправляти прості текстові дані – і на плату Arduino і від неї. Також на платі Leonardo є світлодіоди RX і TX, які спалахують при передачі даних по USB-з'єднанню з комп'ютером (але не по послідовному порту на 0-му і 1-му контактах). Завдяки бібліотеці SoftwareSerial для послідовної комунікації можна використовувати будь-який цифровий контакт Leonardo.

Крім того, Leonardo підтримує комунікацію типу TWI (I2C) та SPI. Для спрощення роботи з інтерфейсом TWI у IDE Arduino є бібліотека Wire; докладніше читайте у документації. Подібну функцію для інтерфейсу SPI виконує бібліотека SPI. Плата Leonardo відображається на комп'ютері як звичайна миша або клавіатура, її можна запрограмувати, щоб керувати цими пристроями за допомогою бібліотек Keyboard і Mouse.

Плата Leonardo програмується за допомогою IDE Arduino.

Захист USB-порту від перевантажень. Плата Leonardo оснащена запобіжником, що самовідновлюється, який захищає USB-порти вашого комп'ютера від коротких замикань і перевантажень. Хоча багато комп'ютерів мають власний захист від подібних неприємностей, запобіжник додає до неї ще один шар. Якщо на USB-порт буде подано більше 500 міліампер, цей запобіжник автоматично перерве з'єднання, доки коротке замикання або перевантаження не буде усунуто.

2.4 Лічильник підрахунку людей на основі інфрачервоного далекоміра GP2Y0A02YK

Інфрачервоний далекомір дозволяє визначати відстань до об'єктів. Це модель GP2Y0A02YK (рис. 2.3) від компанії Sharp. Сенсор визначає відстань відбитого променя світла в інфрачервоному спектрі. Далекімір може використовуватися для обходу завад та орієнтування на місцевості. Виводом є аналоговий сигнал, з рівнем напруги, залежним від відстані до мети у встановленому напрямку.



Рисунок 2.3 – Зовнішній вид інфрачервоного далекоміра GP2Y0A02YK та кабелю до нього

Джерело [27]

Датчик підключається до керуючої електронної частини через 3 дроти. При підключенні до Arduino дуже зручно використовувати Troyka Shield.

Характеристики далекоміра GP2Y0A02YK наступні:

- напруга живлення: 5 В;
- споживаний струм: 33-50 мА;
- діапазон відстаней: 20-150 см.

Оскільки в основі роботи пристрою використовується світло, сенсор погано підходить для визначення відстані до об'єктів світлопоглинання. Далекімір навіть не відчує прозору поверхню, наприклад із пластику або

оргскла. Для визначення відстаней у такому оточенні може бути корисний ультразвуковий далекомір URM37 або HC-SR04.

Цей далекомір має досить велику мертву зону перед собою: 20 см. Якщо необхідно розглядати перешкоди на менших відстанях, а гранична відстань не така важлива, розгляньте далекомір для дистанцій 10-80 см або 4-30 см тієї ж лінійки. Можна досягти більшої гнучкості, комбінуючи датчики з різними діапазонами.

2.5 Мікроконтролер Arduino UNO для управління лічильниками

Платформа Arduino представляє собою відкриту апаратну-програмну платформу, яка дозволяє збирати різні електронні пристрої, і тому вона є корисною для розробників різного обладнання з інформаційної безпеки та керування доступом. При цьому розроблені пристрої та обладнання може працювати як автономному режимі, а також по мережі для взаємодії з сервером для обміну з даними.

Ця платформа складається з апаратної та програмної частин; обидві надзвичайно гнучкі та прості у використанні. Для програмування приладів, що входять до платформи Arduino використовується спрощена версія C++, також відома як Wiring. Розробку програм можна проводити як з використанням безкоштовного середовища Arduino IDE, так і за допомогою графічної мови програмування XOD IDE або довільного інструментарію C/C++. Програмна платформа Arduino підтримуються операційними системи Windows, MacOS X та Linux.

Для програмування та взаємодії з комп'ютером розробнику потрібен USB-кабель. Для автономної роботи буде потрібно блок живлення на 7,5-12В. Мікроконтролер Arduino Uno може житися як від USB-підключення, так і зовнішнього джерела: батареї або мережевого адаптера. Джерело визначається автоматично.

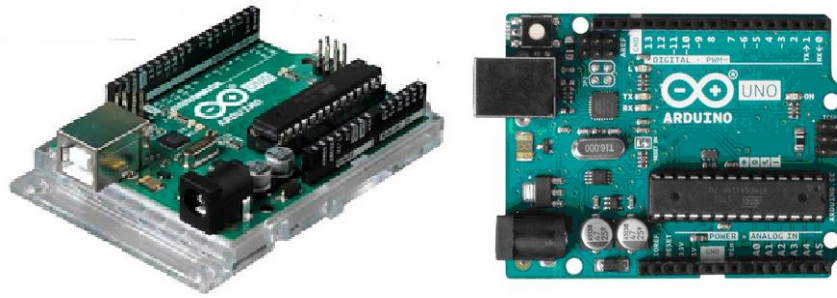


Рисунок 2.4 – Зовнішній вид плати Arduino UNO

Джерело [28]

Платформа може працювати з джерелом напруги від 6 до 20 В. Однак при напрузі менше 7 В робота може бути нестійкою, а напруга більше 12 В може призвести до перегріву та пошкодження. Тому діапазон, що рекомендується, становить 7-12 В. Мікроконтролер Arduino має такі контакти для доступу до джерела живлення:

- контакт Vin надає той самий вольтаж, що використовується для живлення платформи. При підключенні через USB напруга дорівнює 5 В;
- контакт 5V надає 5В незалежно від вхідної напруги. На цій напрузі працює процесор. Максимальний допустимий струм цього контакту складає 800 мА;
- контакт 3.3V надає 3,3 В. Максимальний допустимий струм із цього контакту - 50 мА;
- контакт GND це мінусова клемма, тобто «земля».

Мікроконтролер Arduino оснащений 32 КБ flash-пам'яті, 2 КБ із яких відведено під так званий bootloader. Це дозволяє прошивати Arduino із звичайного комп'ютера через USB. Ця пам'ять постійна і не призначена для зміни в процесі роботи пристрою. Основне її призначення це зберігання програми та супутніх статичних ресурсів.

Мікроконтролер Arduino має також 2Кбайт SRAM-пам'яті, які використовуються для зберігання тимчасових даних на кшталт змінних програм, тобто це фактично оперативна пам'ять плати. SRAM-пам'ять очищується при знеструмленні. Мікроконтролер Arduino має 1 КБ EEPROM-

пам'яті для довготривалого зберігання даних. За своїм призначенням це аналог жорсткого диска Arduino. На платформі розташовано 14 контактів (pins), які можуть бути використані для цифрового введення та виведення. Яку роль виконує кожен контакт залежить від вашої програми. Всі вони працюють з напругою 5 і розраховані на струм до 40 мА. Також кожен контакт має вбудований, але відключений за замовчуванням резистор на 20-50 кОм. Деякі контакти мають додаткові ролі:

- Serial: 0-й та 1-й, використовуються для прийому та передачі даних USB;
- зовнішнє переривання: 2-й та 3-й, ці контакти можуть бути налаштовані, щоб почати виклик заданої функції при зміні вхідного сигналу;
- PWM: 3-й, 5-й, 6-й, 9-й, 10-й та 11-й, можуть бути виходами із широтно-імпульсною модуляцією (pulse-width modulation) з 256 градаціями;
- LED: 13-й, до цього контакту підключений вбудований світлодіод, якщо на контакт виводиться напруга 5В, світлодіод запалюється, а при напрузі 0В він гасне.

Крім контактів цифрового введення-виведення мікроконтролер Arduino має 6 контактів аналогового введення, кожен з яких надає дозвіл у 1024 градації. За замовчуванням значення міряється між землею та 5 В, однак можна змінити верхню межу, подавши напругу необхідної величини на спеціальний контакт AREF. Крім цього, на платі є вхідний контакт Reset. Його встановлення в логічний нуль призводить до скидання процесора. Це аналог кнопки Reset звичайного комп'ютера.

Мікроконтролер Arduino Uno має кілька способів взаємодії з іншими мікроконтролерами Arduino та комп'ютерами. Платформа дозволяє встановити послідовне (Serial UART TTL) з'єднання через контакти 0 (RX) та 1 (TX). Встановлений на платформі чіп ATmega16U2 транслює це з'єднання через USB: на комп'ютері стає доступним віртуальний COM-порт. Програмна частина Arduino включає утиліту, яка дозволяє обмінюватися текстовими повідомленнями цього каналу. Вбудовані в плату світлодіоди RX і TX

світяться, коли передача даних між чіпом ATmega162U і USB-портом комп'ютера. Окрема бібліотека дозволяє організувати послідовне з'єднання з використанням будь-яких інших контактів, не обмежуючись штатними 0-м та 1-м. За допомогою додаткових плат розширення стають доступні інші способи зв'язку: мережа Ethernet, радіоканал, бездротові мережі. Arduino Uno має запобіжник, що захищає USB-порти вашого комп'ютера від перенапруги та коротких замикань. Хоча більшість комп'ютерів мають власні засоби захисту, запобіжник дає додаткову перевагу. Він забезпечує розрив з'єднання, якщо USB-порт подається більше 500 мА.

2.6 Висновок за розділом

Результати аналізу другого розділу дали змогу обґрунтувати вибір обладнання для розробки системи підрахунку кількості людей в кімнатах з метою організації інформаційної безпеки підприємства. Для завдання організації мережі обрано стандарт RS-485, за допомогою якого реалізується передача даних від кожного зчитувача людей кімнати офісу. Стандарт RS-485 забезпечує дальність передачі даних до 1200 метрів, що достатньо для організації мережі малого офісу. Для реалізації стандарту RS-485 обрано плату RS-485 Shield, що дозволяє підключати його по проводам до мікроконтролера ARDUINO UNO. Обрано мікроконтролер ARDUINO UNO для взаємодії з інфрачервоним далекоміром GP2Y0A02YK. Дані, що поступають від далекоміра GP2Y0A02YK, передаються по мережі RS-485. Перевагою далекоміра GP2Y0A02YK є обмежений діапазон виміру відстані і малий кут реєстрації суб'єкта, що зменшує ймовірність випадкових спрацювань від завад.

3 РОЗРОБКА МЕРЕЖІ СИСТЕМИ ПІДРАХУНКУ ЛЮДЕЙ В ПРИМІЩЕННЯХ ОФІСУ

3.1 Основні відомості про план розташування приміщень офісу

Розглянемо наступний типовий план приміщення офісу, який характерний для офісів малого бізнесу та був зроблений на основі аналітичного аналізу підприємств. Як правило, для офісів малого бізнесу характерні невеликі приміщення з 3-5 кімнат, що насамперед пов'язано з великою орендною платою. Характерні наступні підрозділи малого офісу, що розміщуються в окремих кімнатах: директор; бухгалтерія; відділ збуту, відділ постачання; контрольно-пропускний пункт (КПП), де розміщується охоронець. Таким чином, офіс буде складатися з 6 кімнат, коридора та холу для очікування відвідувачів. На рис. 3.1 надано типовий план приміщень офісу де необхідно розташувати обладнання для підрахунку людей:

- 1) шість кімнат: 25 кв м – директора; 30 кв м – бухгалтерія; 25 кв м – відділ постачання; 30 кв м – відділ збуту; 35 кв м – КПП; 7 кв м – санітарне приміщення (СП);
- 2) коридор – 26 м кв;
- 3) хол – 30 м кв.

3.2 Вибір типу лічильників та способу підключення

Розглянемо вибір типу лічильника, який працює на основі інфрачервоного променю. Такі лічильники бувають декількох типів – вертикальні (рис. 3.2) та горизонтальні (рис. 3.3).

Лічильники підрахунку людей працюють за принципом переривання інфрачервоного променю. Обов'язковими приладами цих лічильників є приймач та передавач. Передавач випромінює інфрачервоний промінь, а приймач приймає. Після перетину променю приймач аналізує факт проходження. Їх, як правило, ділять на два види – із підключенням до персонального

комп'ютера (ПК) чи ні. У першому варіанті дані виводяться на ПК, у другому на дисплей лічильника.

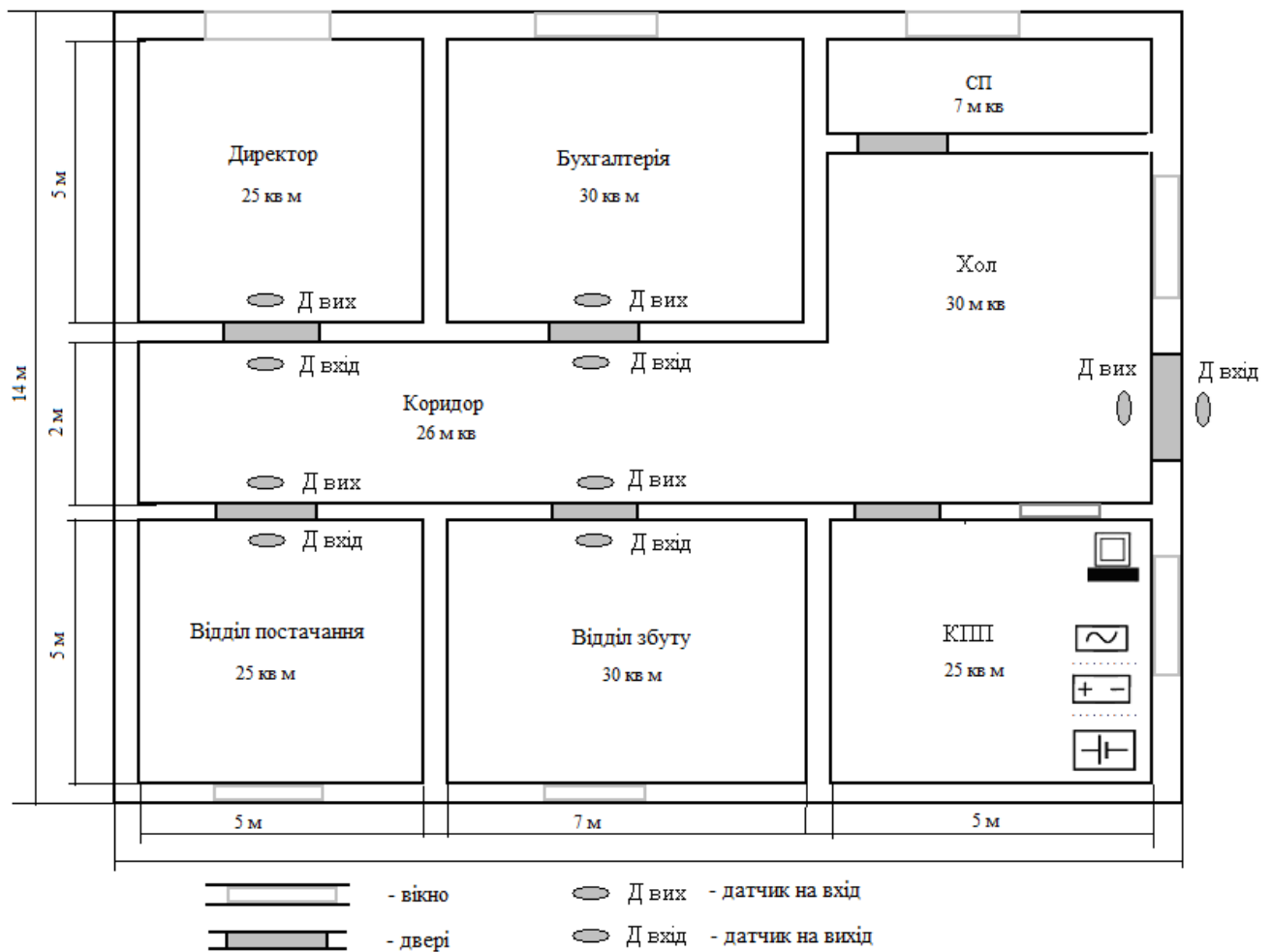


Рисунок 3.1 – План приміщень офісу з розташуванням обладнання для підрахунку людей в кімнатах офісу

Джерело: авторська розробка

Розглянемо варіанти підключення лічильників до джерела живлення:

1) автономне живлення – на батарейках – лічильники можуть мати автономне живлення, бездротовий лічильник означає фізичне середовище зв'язку з ПК радіо або GSM, двонаправлений або однонаправлений – розрізняє напрямки руху чи ні, з архівом даних – має вбудовану пам'ять і годинник реального часу;

3) живлення від центрального джерела.

Клієнт визначає необхідні типи для зручнішого монтажу. Наприклад, лічильник з підключенням до ПК, на батарейках, бездротовий. Значить робота з даними виконується на ПК, а лічильник двостороннім скотчем клеїться на проході.

Лічильники людей, які підключаються до ПК, піддаються простий автоматизації збору даних для аналізу. У простому варіанті дані проглядаються на локальному ПК. Коли є мережа торгових точок, дані передаються через Інтернет в центральний офіс з усіх точок. У центральному офісі дані проглядаються на сервері з будь-якого ПК або на сервері за допомогою ПЗ. Сервер – це ПК, до якого надходять данні. Якщо сервер це ПК виробника лічильників, переглядати дані можна через веб браузер з будь-якої точки землі.

Переваги: недорогі; висока точність; прості в установці; точність на проході до 1,5 м – 99% (100 люд/год).

Недоліки

- похибка зростає при збільшенні потоку та зростанні ширини проходу, оскільки зростає можливість проходу на одній лінії двох людей. Похибка підрахунку в окремих випадках може досягати 20% [10].

- відсутність можливості контролю за коректною роботою обладнання (продавці фізично закривають лічильники, щоб знизити кількість відвідувачів та підвищити коефіцієнт продажу).

Вертикальні лічильники (рис. 3.2) працюють за принципом відображення променю тіла людини.

Теоретично вертикальні промені точніші, ніж горизонтальні, і разі застосування сучасних моделей можна досягти високої точності підрахунку відвідувачів до 95 %. Однак, це справедливо не для всіх модифікацій. Більшість вертикальних інфрачервоних сенсорів є «рампу» встановлену над проходом, в якій на відстані близько 50-60 см розташовуються вертикальні інфрачервоні сенсори.



Рисунок 3.2 – Вертикальний інфрачервоний лічильник для підрахунку людей в кімнатах

Джерело: авторська розробка

Алгоритм простий – перетин одного променя дорівнює одному відвідувачеві. У зв'язку з цим такі системи можуть давати дуже високі похибки, так як ширина плечей людини коливається в діапазоні 35-70 см і більше (взимку).

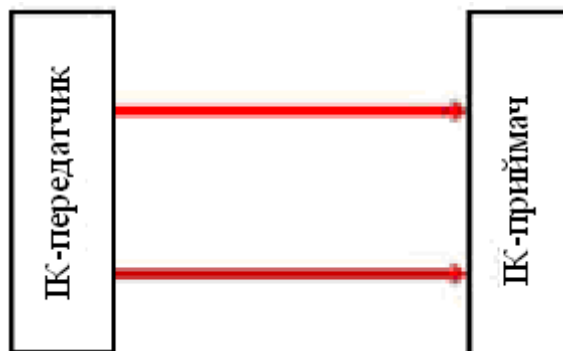


Рисунок 3.3 – Горизонтальний інфрачервоний лічильник для підрахунку людей в кімнаті

Джерело: авторська розробка

Як правило, більшість горизонтальних і вертикальних інфрачервоних сенсорів дозволяє вести статистику відвідуваності на основі ПК користувача. Основні недоліки: висока ціна; низька точність при щільному потоці.

На основі зробленого аналізу визначимо:

- 1) для невеликого офісу доцільно застосувати вертикальні лічильники, які працюють за принципом відображення променя тіла людини;
- 2) доцільне застосування двонаправлених вертикальних лічильників, які більш точніші, ніж горизонтальні;
- 3) двонаправлені вертикальні лічильники, встановлюється на проході.
- 4) централізоване живлення кожного лічильника;
- 5) відображення результатів підрахунків на сервері.

На рис. 3.4 надано розташування і принцип дії вертикальних інфрачервоних лічильників, які розміщуються над отвором приміщення.

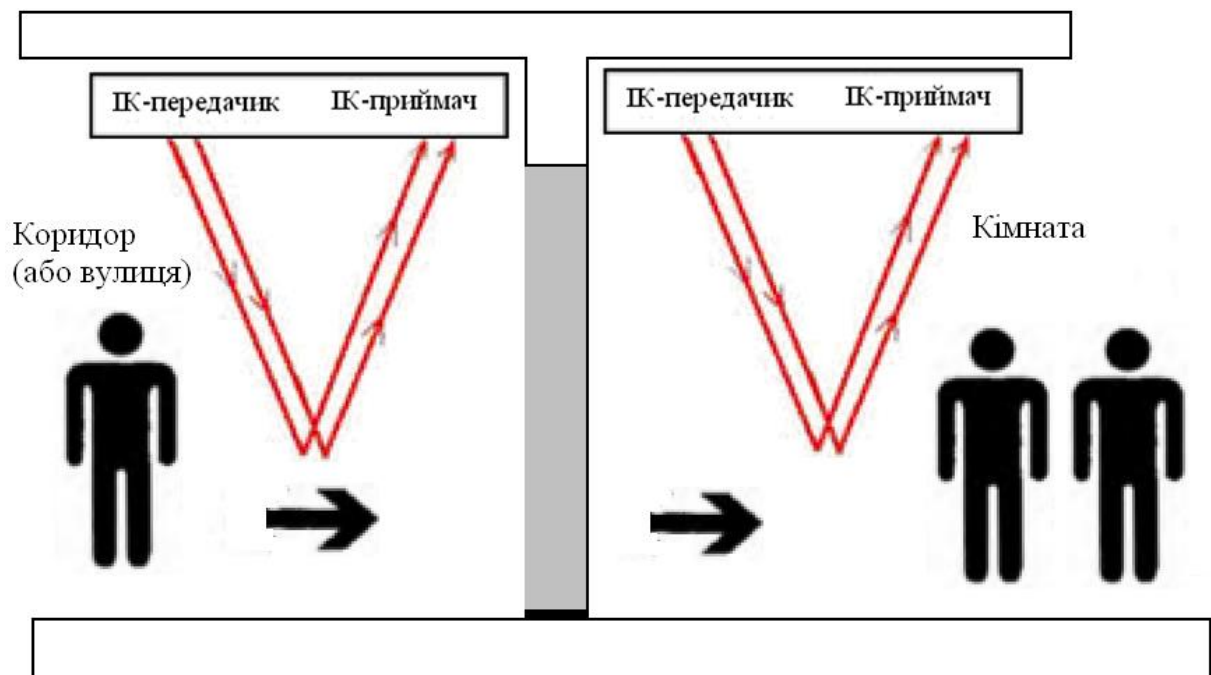


Рисунок 3.4 – Вертикальні інфрачервоні лічильники для підрахунку людей в кімнаті

Джерело: авторська розробка

3.3 Комплектування та монтаж обладнання підрахунку людей в приміщеннях офісу

Розглянемо склад обладнання, яке буде керуючим пристроєм для керування мережею лічильників підрахунку людей в приміщенні офісу:

- 1) мікроконтролер Arduino Leonardo – 1 шт;
- 2) DF Robot RS485-шилд – 1 шт;
- 3) імпульсний блок живлення – 1 шт;
- 4) кабель Micro USB – 1 шт;
- 5) сервер (ПК) – 1 шт.

Для монтажу керуючого обладнання необхідно виконати наступні дії:

- 1) встановіть RS485-шилд на Arduino Leonardo;
- 2) підключіть імпульсне джерело живлення до Arduino Leonardo;
- 3) підключити лінію «А» та «В» мережі RS485 до клемника;
- 4) підключіть лінії GND і VIN мережі RS485 до відповідних висновків Arduino;
- 5) підключіть USB-кабелем плату Arduino до ПК.

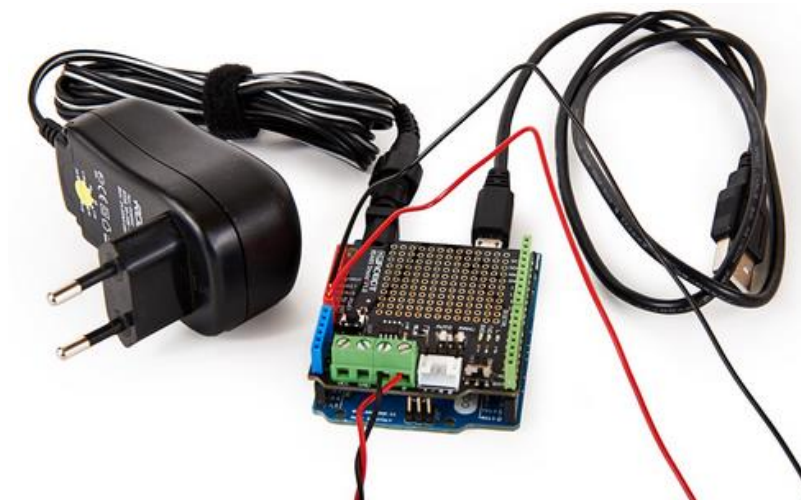


Рисунок 3.5 – Підключення керуючого пристрою на основі мікроконтролер Arduino Leonardo та DF Robot RS485-шилд

Джерело [29]

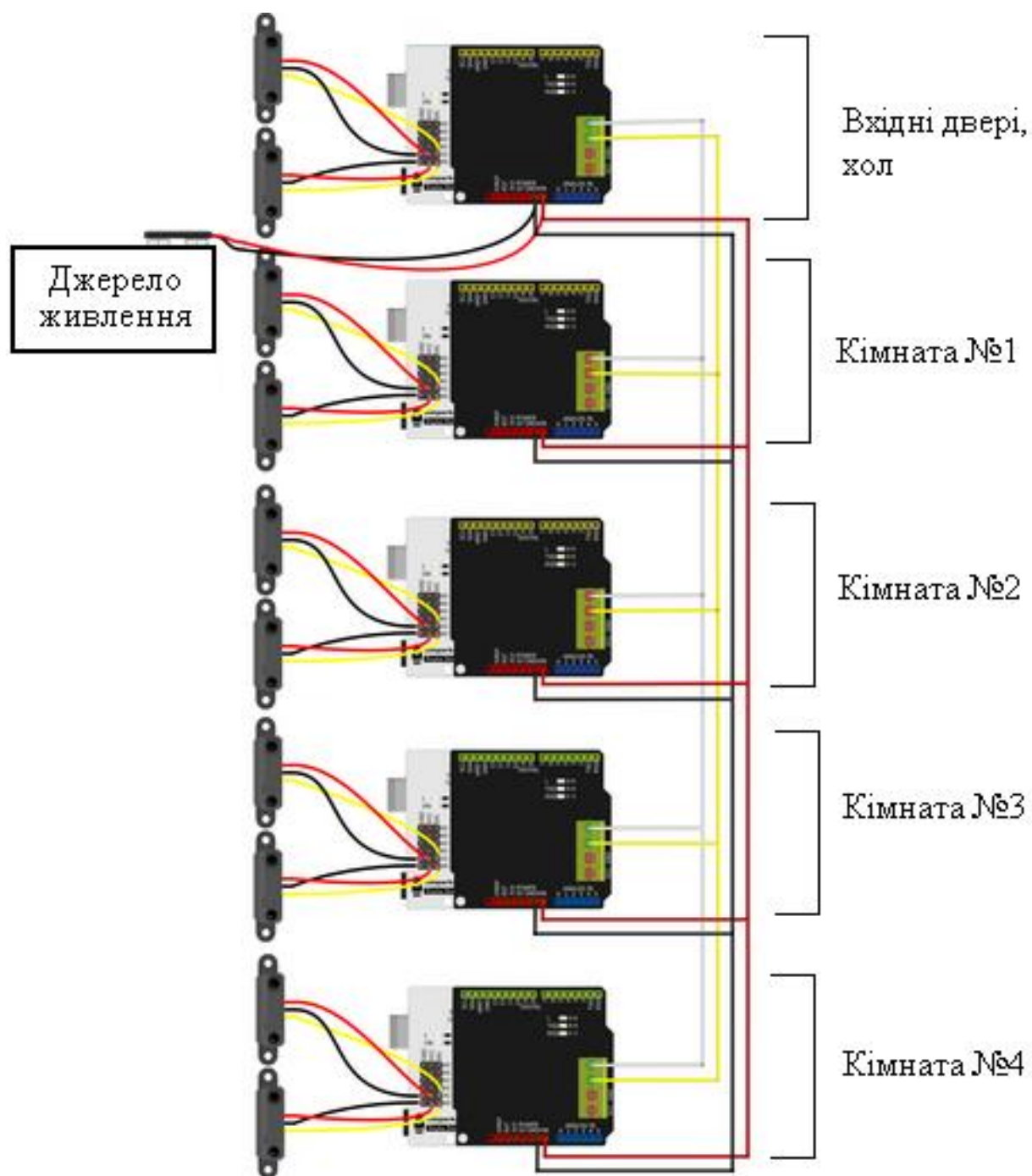


Рисунок 3.6 – Підключення відомих пристроїв-лічильників в мережі на основі Arduino Uno

Джерело: авторська розробка

Розглянемо склад обладнання (одного відомого пристрою), яке буде керувати лічильниками в мережі та передавати данні по мережі RS485-шилд на головний мікроконтролер Arduino Leonardo:

- 1) Arduino Uno – 1 шт;
- 2) DFRobot RS485-шилд – 1 шт;
- 3) Breadboard Mini – 1 шт;
- 4) сполучні дроти – 10 шт;
- 5) інфрачервоний далекомір Sharp (20-150 см) × 2 шт;
- 6) мідний провід перетином 0,5-1 мм для прокладання мережі.

Для монтажу відомого пристрою потрібно виконати наступне:

- 1) встановити Troyka-шилд на Arduino Uno;
- 2) встановити RS485-шилд на Troyka-шилд;
- 3) підключити далекоміри до аналогових клем A0 та A1;
- 4) підключити лінії «А» та «В» мережі RS485 до відповідних клем;
- 5) підключити лінії GND і VIN мережі RS485 до відповідних виводів

Arduino UNO.

3.4 Алгоритм роботи мережі та розробка програмного забезпечення

При проходженні людини через дверний отвір, він спочатку потрапляє під промені одного датчика, а потім другого. По тому, який датчик спрацював першим, можна визначити напрямок перетину дверного отвору. Емпіричні напруги лічильників при проходженні людини через отвір в залежності від часу надано на рис. 3.7.

Рівень напруги, що вище за V_{ref} , сприймається як одиницю, а решта – як нулі. V_{ref} треба підбирати виходячи з розмірів дверних косяків та параметрів датчиків.

Кожен вузол зберігає кількість проходів. При виявленні перетину дверного отвору пристрій додає "1" до змінної, якщо людина пройшла в один бік, і віднімає "1" у протилежному випадку.

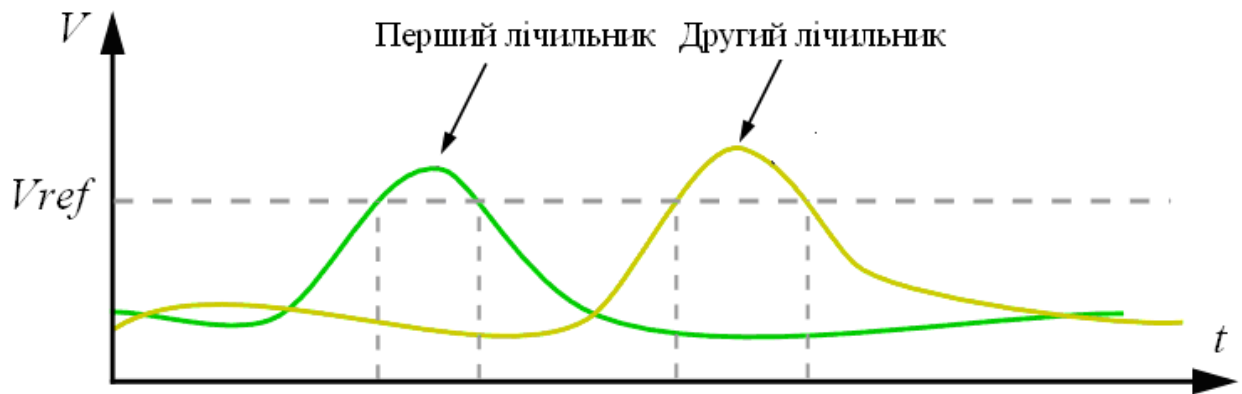


Рисунок 3.7 – Епюри напруги лічильників при проходженні людини через отвір в залежності від часу

Джерело: авторська розробка

Провідний пристрій циклічно опитує всі вузли, надсилаючи пакет із номером пристрою. Ці пакети отримують всі відомі пристрої одночасно, однак відповідає тільки той пристрій, номер якого вказаний у пакеті. Номер пристрою задається висновками "8", "9", "10".

У пакеті у відповідь пристрій повідомляє ведучому пристрою значення збереженої змінної проходів. Після цього пристрій обнулює це значення. Якщо між двома опитуваннями одного пристрою було два проходи в різні боки, ці події можуть «вислизнути» від ведучого. Щоб цього не сталося, опитування проводиться дуже часто - десятки разів на секунду.

Керовані пристрої ніколи не починають передачу за власним бажанням. Вони лише відповідають на запити ведучого. Цим вирішується проблема колізій на лінії.

Провідний пристрій заснований на Arduino Leonardo. Це дозволяє використовувати одночасно RS485-шилд та спілкування по Serial між Arduino та ПК. Дані, зібрані з усіх ведених пристроїв, передаються по Serial, де їх отримує Processing-програма та відображає у графічному вигляді. Програма повинна малювати план приміщень і відображає в кожній кімнаті кількість людей, які там знаходяться. Розроблена програма на мові C++ надана у додатку Б.

Розробка власних програм на базі плат, сумісних з архітектурою Arduino, здійснюється в офіційному безкоштовному середовищі програмування Arduino IDE. Середовище налаштовано для написання, компіляції та завантаження власних програм в пам'ять мікроконтролера, встановленого на платі Arduino-сумісного пристрою. Основою середовища розробки є мова Processing/Wiring – це фактично звичайний C++, доповнений простими і зрозумілими функціями для керування введенням/виводом даних на контактах. Існують версії середовища для операційних систем Windows, Mac OS і Linux.

Процес установки середі програмування Arduino IDE на комп'ютері з операційною системою Windows полягає в наступному.

Для інсталяції програмного забезпечення Arduino IDE на комп'ютері з операційною системою Windows цього потрібно зайти на веб-сторінку www.arduino.cc, вибирати версію для операційної системи Windows і скачати архівний файл. Цей файл має все необхідне і, в тому числі, потрібні драйвери на пристрої. Після закінчення завантаження потрібно розпакувати файл на жорсткому диску ПЕВМ.

Далі необхідно встановити драйвери. Підключаємо Arduino до комп'ютера. На контролері повинен засвітитися зелений світлодіод, який показує, що живлення до нього подається. ОС Windows починає спробу встановити драйвер, яка закінчується повідомленням «Можливо, драйвер не було встановлено». Далі потрібно відкрити «Диспетчер пристроїв». У складі пристроїв знаходимо значок «Arduino Uno» – пристрій відзначено знаком оклику. Клацаємо правою кнопкою миші на значку «Arduino Uno» і у вікні, вибираємо пункт «Оновити драйвери» і далі пункт «Виконати пошук драйверів» на цьому комп'ютері. Вказуємо шлях до драйвера – ту папку на комп'ютері, куди витягли скачаний архів. Нехай це буде папка drivers каталогу установки Arduino, наприклад, C:\arduino-1.0\drivers. Ігноруємо все попередження Windows і отримуємо в результаті повідомленням – «оновлення програмного забезпечення для даного пристрою завершено успішно». У

заголовку вікна буде вказано COM-порт, на який встановлено пристрій. Далі можна запускати Arduino IDE.

Середовище розробки Arduino (рис. 3.8) має: редактора програмного коду; б) області повідомлень; вікна виведення тексту; панелі інструментів з кнопками часто використовуваних команд; декількох меню.

Програма, що написана в середовищі Arduino, носить назву скетч. Скетч пишеться в будованому текстовому редакторі, який має колірне підсвічування створюваного програмного коду. Під час збереження і експорту проекту в області повідомлень з'являються пояснення і інформація про помилки. Вікно виведення тексту показує повідомлення Arduino, що включають повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити і записати програму, створити, відкрити і зберегти скетч, відкрити моніторинг послідовної шини.

Програмісти, що розробляють скетчі, мають можливість застосування додаткових функцій з бібліотек, які представляють собою спеціальним чином оформлений програмний код.

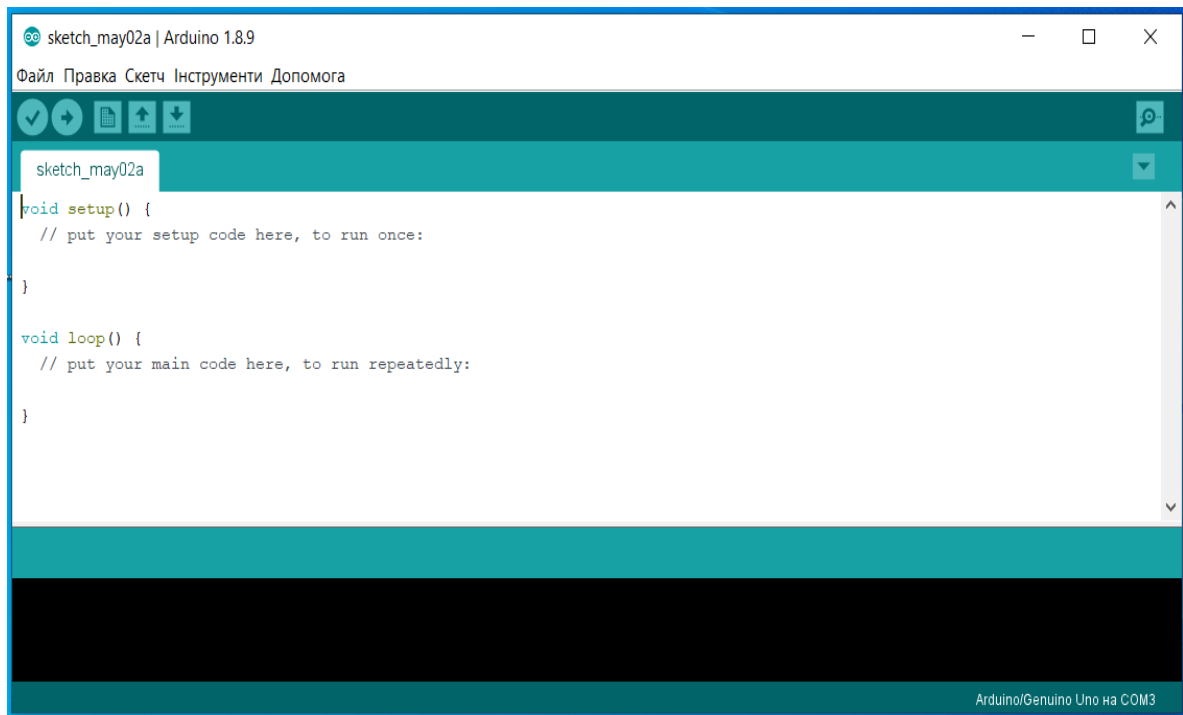


Рисунок 3.8 – Програмний інтерфейс середовища розробки Arduino

Джерело: авторська розробка

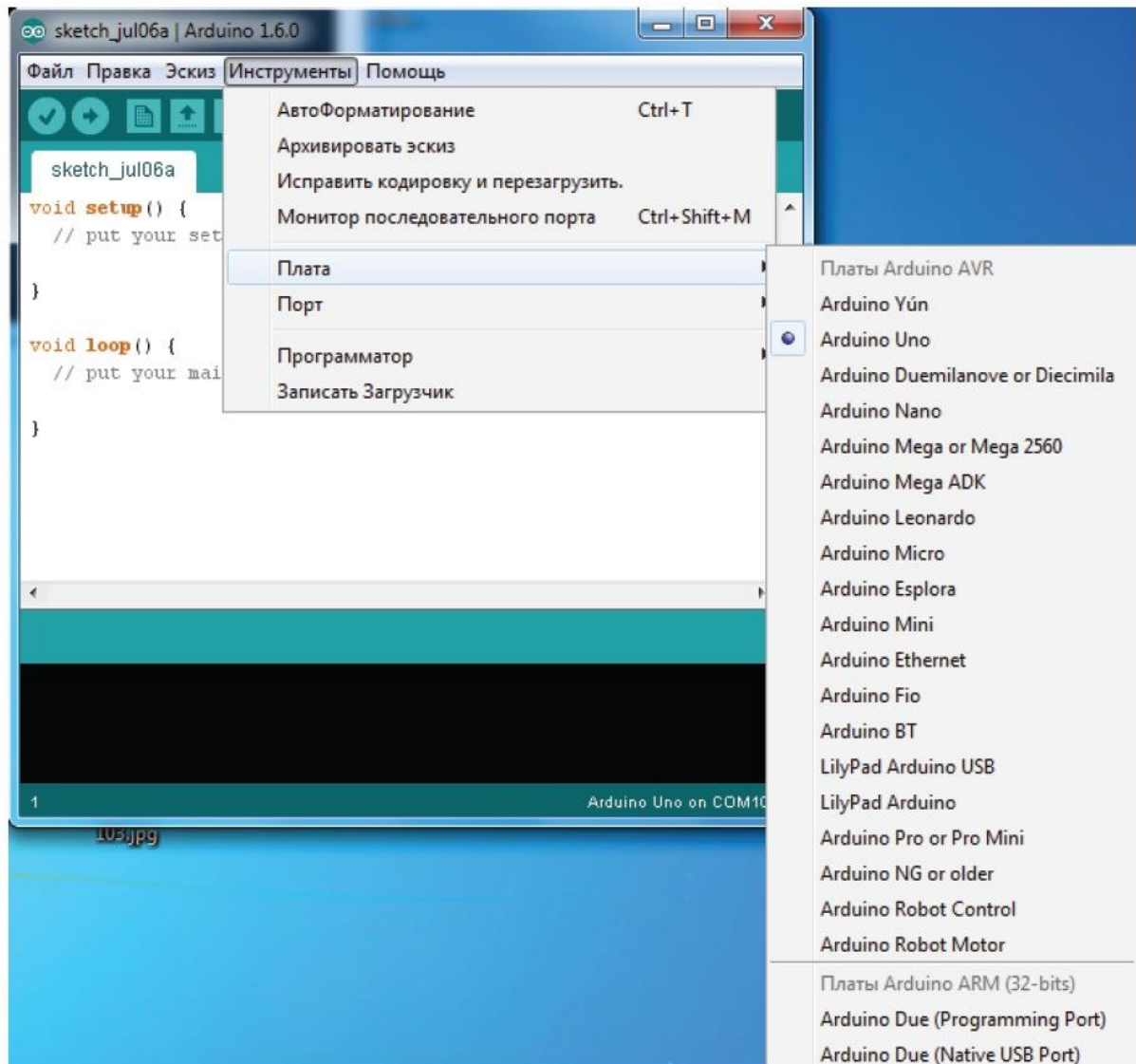


Рисунок 3.9 – Приклад вибору Arduino плати

Джерело: авторська розробка

За допомогою цих функцій, як правило, реалізовується деякий функціонал, який можна підключити до створюваного проекту. Характерно існування великої кількості спеціалізованих бібліотек. Зазвичай ці бібліотеки пишуться так, щоб спростити вирішення тієї чи іншої задачі і приховати від розробника деталі програмно-апаратної реалізації. Серед Arduino IDE поставляється з набором стандартних бібліотек. Вони знаходяться в підкаталозі `libraries` каталогу установки Arduino. Необхідні бібліотеки можуть бути також завантажені з різних ресурсів. Якщо бібліотека встановлена правильно, то вона з'являється в меню «Ескіз» – «Імпорт бібліотек». Вибір

бібліотеки в меню призведе до додавання до початкового коду рядка: `#include <ім'я бібліотеки.h>`.

Ця директива підключає заголовки з описом об'єктів, функцій і констант бібліотеки, які тепер можуть бути використані в проекті. Серед Arduino буде компілювати створюваний проект разом із зазначеною бібліотекою.

Перед завантаженням скетчу потрібно задати необхідні параметри в меню «Інструменти» – «Плата (Tools / Board)» (рис. 3.9) і «Поліграфія» – «Послідовний порт» (рис. 3.10).

Сучасні платформи Arduino перезавантажуються автоматично перед завантаженням. На старих платформах необхідно натиснути кнопку перезавантаження. На більшості плат під час процесу завантаження будуть мигати світлодіоди RX і TX. При завантаженні скетчу використовується завантажувач (bootloader) Arduino – невелика програма, що завантажується в мікроконтролер на платі. Вона дозволяє завантажувати програмний код без використання додаткових апаратних засобів. Робота завантажувача визначається по миганню світлодіода на цифровому виводі D13.

Монітор послідовного порту (Serial Monitor) відображає дані, що їх посилають в платформу Arduino (плату з інтерфейсом USB або послідовної шини).

Слід зазначити, що зараз також випускаються моделі мікроконтролерів Arduino, в яких в якості USB-чіпсета використовуються мікросхеми серії CH340 (рис. 3.11).

Таким чином, можливості платформи Arduino можуть бути гідні для реалізації проекту відповідно до завдання даної роботи. Встановлено, що платформа Arduino має відповідну середу розробки, тобто елементну базу та програмне забезпечення. Це забезпечує швидку розробку електронних пристроїв. Платформа Arduino має просту мови програмування та відкриту архітектуру. Плата Arduino складається з мікроконтролера Atmel AVR і елементів обв'язки для програмування та інтеграції з іншими схемами.

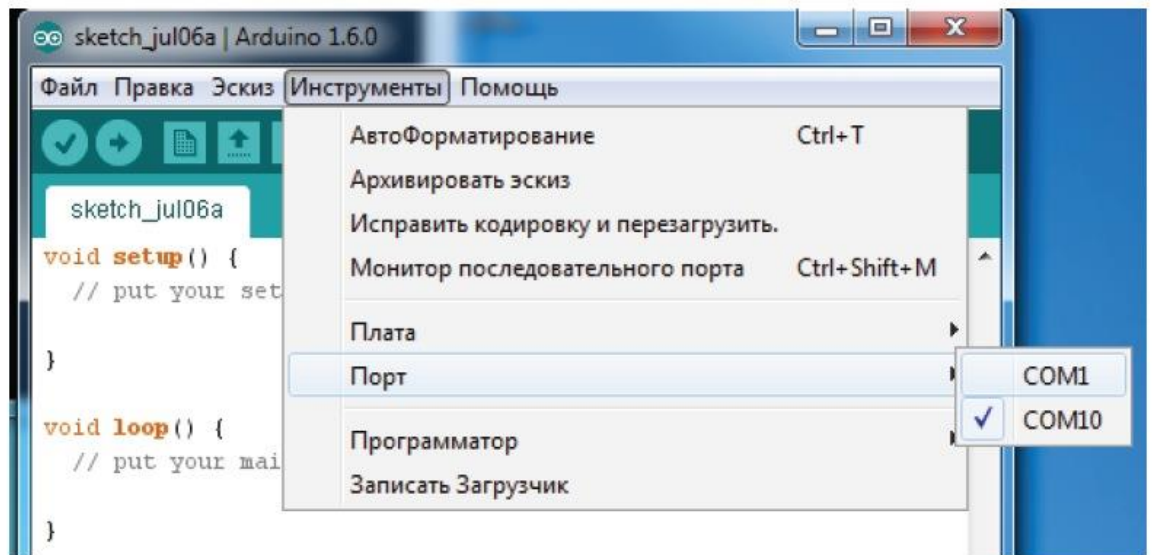


Рисунок 3.10 – Приклад вибору порту підключення плати Arduino

Джерело: авторська розробка

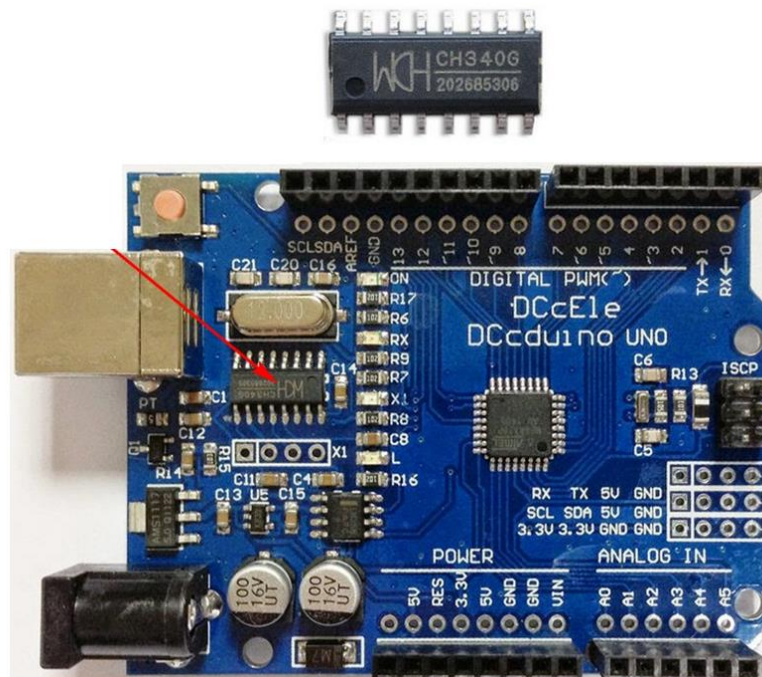


Рисунок 3.11 – Модель мікроконтролера Arduino, в яких в якості USB-чіпсета використовуються мікросхеми серії CH340

Джерело [30]

3.5 Блок схеми програмного забезпечення системи підрахунку кількості людей в приміщеннях

Виходячи з опису та алгоритму системи підрахунку кількості людей в приміщеннях офісу розроблено програму на мові C ++ в середовищі програмування Arduino IDE із застосуваннями відповідних бібліотечних функцій платформи Arduino. Блок схеми програмного забезпечення системи підрахунку кількості людей в приміщеннях надано на рис. 3.12.

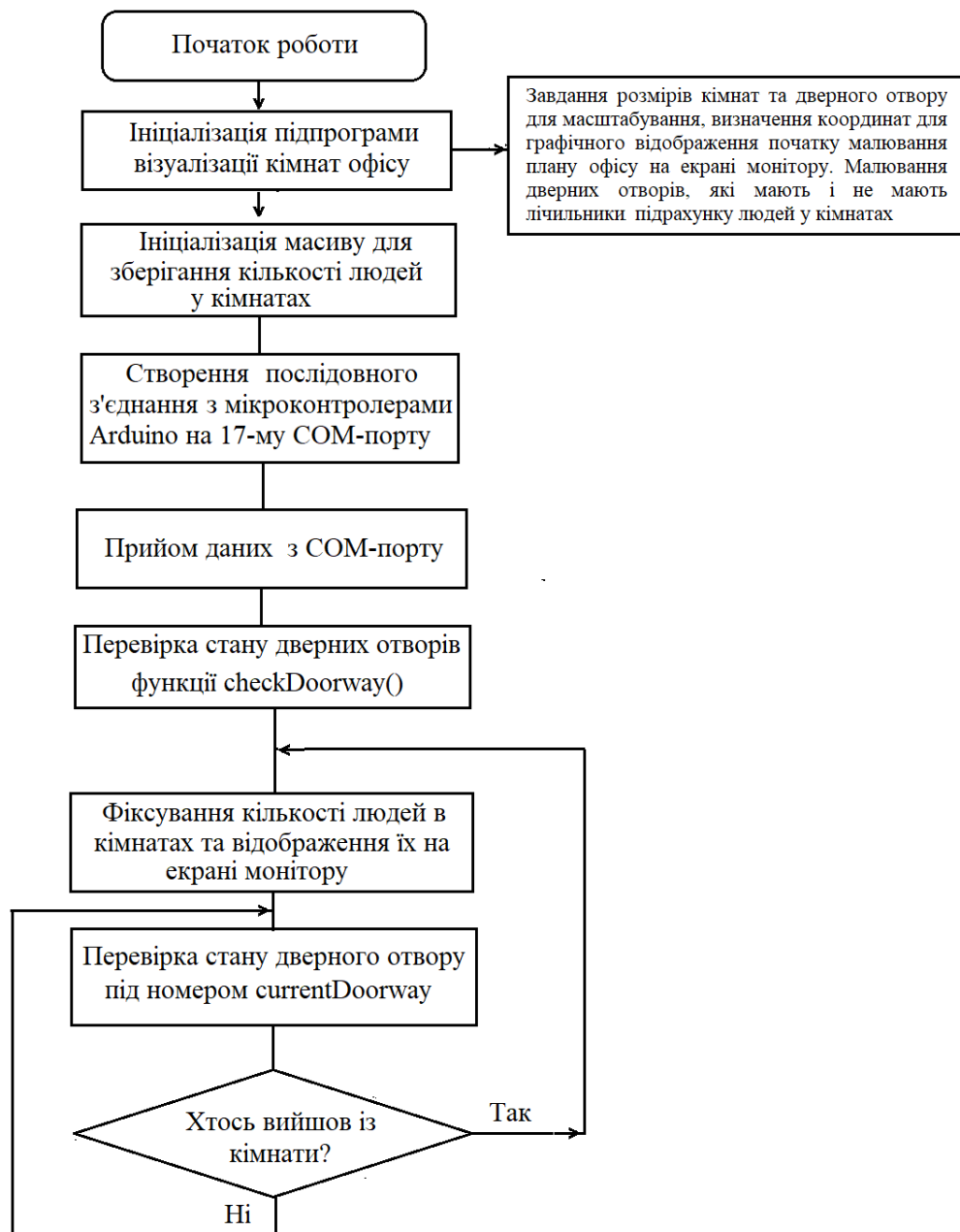


Рисунок 3.12 – Блок схеми програмного забезпечення системи підрахунку кількості людей в приміщеннях офісу

Блок схема програмного забезпечення складається з наступних модулів:

- 1) Початок роботи системи, тобто підключення до джерела живлення;
- 2) Ініціалізація підпрограми візуалізації кімнат офісу:

- `import processing.serial.*` та `Serial port` - завдається можливість роботи з послідовним портом.

- `int roomScale = 3` – задаються розміри кімнат та дверного отвору для масштабування;

- `int drawStart = 75` – визначаються координати для графічного відображення початку малювання плану офісу на екрані монітору;

- `int doorwayWidth = 15 * roomScale` – задається ширина дверного отвору;

- `int[] roomWidth = {`

`130 * roomScale, 60 * roomScale, 25 * roomScale, 85 * roomScale, 130 *`

`roomScale` – задається масив ширини кімнат;

- `Integer[] persons = { 0, 0, 0, 0 }` – масив для зберігання кількості людей в кімнатах офісу;

- `int[] doorwayState = { 0, 0, 0, 0 }` – поточний стан кожного отвору;

- `void setup()` - функція виконується один раз при старті скетчу, за допомогою якої встановлюються початкові параметрів різних підфункцій;

- 3) Ініціалізація масиву для зберігання кількості людей у кімнатах

- `if (checkDoorway(0)) {`

`// дверний отвір буде червоним`

`stroke(255, 0, 0);`

`} else {`

`// інакше - темно-сірий`

`stroke(50);`

`} - перевірка стану дверних отворів функції checkDoorway(), якщо`

`хтось пройшов;`

- `line(drawStart + roomWidth[0],`

`drawStart + roomHeight[0] * 0.6,`

`drawStart + roomWidth[0],`

```

        doorwayWidth + drawStart + roomHeight[0] * 0.6); - малювання
отвору;
-    text(persons[0].toString(), drawStart + roomWidth[0]/2,
        drawStart + roomHeight[0]/2);
    text(persons[1].toString(), drawStart + roomWidth[0]+roomWidth[1]/2,
        drawStart + roomHeight[1]/2); - центрах плану кімнат вказується
кількість осіб у кімнаті fill(255);
- boolean checkDoorway(int currentDoorway) - перевірка стану дверного
отвору під номером currentDoorway;
{
    // Якщо ніхто не проходив – повернемо false
    boolean result = false;
    switch(doorwayState[currentDoorway]) {
        // Якщо хтось вийшов із кімнати
        case 1:
            result = true; // стан змінився
            // якщо у кімнаті було більше 0 осіб
            if (persons[currentDoorway] > 0) {
                // заберемо одну людину з цієї кімнати
                persons[currentDoorway]--;
            }
            // додавання його до сусідньої кімнати
            persons[currentDoorway+1]++;
            // зміна стану відпрацьовано
            doorwayState[currentDoorway] = 0;
            break;
        // Якщо хтось увійшов до кімнати
        case 2:
            result = true; // состояние изменилось
            // якщо у сусідній кімнаті більше 0 осіб

```

```

if (persons[currentDoorway+1] > 0) {
    // заберемо одну людину із сусідньої кімнати
    persons[currentDoorway+1]--;
}
// і додамо до іншої кімнати
persons[currentDoorway]++;
// зміна стану і фіксація його в пам'яті
doorwayState[currentDoorway] = 0;
break;
}
// Повертаємо стан дверного отвору
return result;

```

4) Створення послідовного з'єднання з мікроконтролерами Arduino на 17-му COM-порту:

void serialEvent (Serial port) - функція отримання байта із послідовного порту

```

{

```

```

    try {

```

5) Прийом даних з COM-порту

// Читання даних від Arduino

String data = port.readStringUntil('\n');

// формат посилки x:y, де x-номер дверного отвору,

// y – його статус. Розбиваємо рядок data на кілька рядків,

// використовуючи ':' як роздільник

String[] list = split(data, ':');

// номер отвору міститься у першому рядку. Відразу переводимо рядок у число

Integer doorwayNumber = int(list[0]);

```

// статус отвору - у другому рядку. Переклад рядок у число
// HARDCODE! substring(0, 1) - копіювання підрядки розміром 1
символ з рядка,
// початок з 0-го символу
Integer state = int(list[1].substring(0, 1));
// Зберігаємо стан дверного отвору
doorwayState[doorwayNumber] = state;
// запис у консоль, що надійшла інформацію для налагодження
println(data);
println(list[0]);
println(list[1]);
}

```

6) Перевірка стану дверних отворів функції checkDoorway()

```

if (checkDoorway(0)) {
    // дверний отвір буде червоним
    stroke(255, 0, 0);
} else {
    // інакше - темно-сірий
    stroke(50);
}

```

7) Фіксування кількості людей в кімнатах офісу та відображення їх на екрані монітору:

```

boolean checkDoorway(int currentDoorway)
{
    // Якщо ніхто не проходив – повернемо false
    boolean result = false;
    switch(doorwayState[currentDoorway]) {
        // Якщо хтось вийшов із кімнати
        case 1:
            result = true; // стан змінився
    }
}

```



```

// якщо у кімнаті було більше 0 осіб
if (persons[currentDoorway] > 0) {
    // заберемо одну людину з цієї кімнати
    persons[currentDoorway]--;
}

// додавання його до сусідньої кімнати
persons[currentDoorway+1]++;

// зміна стану відпрацьовано
doorwayState[currentDoorway] = 0;
break;

// Якщо хтось увійшов до кімнати
case 2:
    result = true; // состояние изменилось
    // якщо у сусідній кімнаті більше 0 осіб
    if (persons[currentDoorway+1] > 0) {
        // заберемо одну людину із сусідньої кімнати
        persons[currentDoorway+1]--;
    }

    // і додамо до іншої кімнати
    persons[currentDoorway]++;

    // зміна стану відпрацьована, запам'ятаємо це
    doorwayState[currentDoorway] = 0;
    break;
}

// Повертаємо стан дверного отвору
return result;
}

```

Таким чином, розроблена програма забезпечує ініціалізацію підпрограми візуалізації кімнат офісу. При цьому задаються розміри кімнат та дверного отвору для масштабування, визначення координат для графічного

відображення початку малювання плану офісу на екрані монітору. Програма відображає на екрані дверні отвори, які мають і не мають лічильники підрахунку людей у кімнатах. Програма формує масив для зберігання кількості людей у кімнатах офісу. При цьому створюється можливість послідовного з'єднання мікроконтролера Arduino Leonardo з мікроконтролерами Arduino за допомогою COM-порту. Прийом даних значень лічильників відбувається з COM-порту. Програма періодично перевіряє стани дверних отворів за допомогою функції `checkDoorway()`. При наявності зміни стану якогось лічильника його значення фіксується і передається по COM порту на основі мікроконтролер Arduino Leonardo, який підключений до комп'ютера. Значення лічильника кімнати запам'ятовується і відображається на екрані монітора комп'ютера. Результати компіляції програми в середовищі програмування Arduino IDE надано на рис. 3.13.



```

sketch_jun06a | Arduino 1.8.9
Файл Правка Скетч Інструменти Допомога

sketch_jun06a $
// Робота з послідовним портом
import processing.serial.*;
Serial port;

// Множник розмірів кімнат для масштабування розмірів
int roomScale = 3;

// Координата початку малювання з X та Y
int drawStart = 75;

// ширина дверного отвару
int doorwayWidth = 15 * roomScale;

// масив ширин кімнат
int[] roomWidth = {
  130 * roomScale, 60 * roomScale, 25 * roomScale, 85 * roomScale, 130 * roomScale
};
// масив висот кімнат
int[] roomHeight = {
  100 * roomScale, 50 * roomScale, 20 * roomScale, 25 * roomScale, 75 * roomScale
};

Результати компіляції:
Скетч використовує 5168 байтів (16%) місця зберігання для програм. Межа 30720 байтів.
Глобальні змінні використовують 373 байтів (18%) динамічної пам'яті, залишаючи 1675 байтів для локальних змінних.

```

Рисунок 3.13 – Результати компіляції програми в середовищі програмування Arduino IDE

3.5 Висновок за розділом

Результати третього розділу є розробка схеми та програмного забезпечення системи підрахунку людей на основі інфрачервоних лічильників.

Обґрунтовано вибір типу лічильника та спосіб його відключення. Зроблено комплектування та схему монтажу обладнання системи підрахунку людей на основі інфрачервоних лічильників. Розроблено алгоритм роботи мережі лічильників підрахунку людей в кімнатах офісу та розроблено відповідне програмне забезпечення на основі платформи Arduino на мові програмування C++.

ВИСНОВКИ

В бакалаврській роботі розроблено мережну систему керуванням доступом, яка призначена для підраховування кількості людей в кімнатах офісу. В результаті виконання роботи доведено можливість створення мережної системи підрахунку людей в кімнатах офісу на основі мікроконтролера Arduino Leonardo та Arduino UNO, які будуть використовуватися як пристрої управління лічильниками, а для взаємодії з периферійними пристроями застосовується протокол обміну даними RS485-шилд, який є бібліотечною функцією платформи Arduino.

Проведено аналіз основних моделей побудови програмно-апаратної системи та мережі контролю доступу на основі інфрачервоних датчиків. Розроблено структурну схему мережі системи лічильників контролю кількості присутніх в кожній кімнаті офісу. Обґрунтована вибір елементної бази та розроблено принципіальну електричну схему мережі лічильників на основі платформи Arduino.

ПЕРЕЛІК ПОСИЛАНЬ

1. 1. Гладких А.А. Базовые принципы информационной безопасности вычислительных сетей учебное пособие для студентов, обучающихся по специальностям 08050565 21040665, 22050165, 23040165 / А.А. Гладких, В.Е. Дементьев; – Ульяновск УлГТУ, 2009, – 156 с. – ISBN 5-89146-344-X
2. 2. Галкин П.В. Исследование дальности и скорости передачи данных по витой паре в промышленных сетях RS-485 и PROFIBUS / П.В. Галкин, В.В. Гавриленко, А.И. Монько // Проблемы телекоммуникаций. - 2016. - № 2.- С. 94-110.
3. 3. Духан Е.И. Применение программно-аппаратных средств защиты компьютерной информации: учебное пособие / Е.И. Духан, И.И. Синадский, Д.А. Хорьков; науч. ред. д-р техн. наук, проф. И.А. Гайдамакин. Екатеринбург: УГТУ-УПИ, 2008, 182 с.
4. 4. Богуш В.М. Інформаційна безпека держави. Навчальний посібник / В.М. Богуш, О.К. Юдін // – К.: «МК-Прес», - 2005. – 432 с.
5. 5. Шон Харрис CISSP Руководство для подготовки к экзамену / Шон Харрис // Пятая редакции, 2019. - 875 с.
6. 6. Вильям Столлингс Криптографическая защита сетей / Вильям Столлингс // – М.: Издательский дом “Вильямс”, 2001.
7. 7 Биометрическая система контроля и управления доступом. [Электроний ресурс]. – Режим доступа: <https://vashtvmir.ru/biometricheskaya-sistema-kontrolya-i-upravleniya-dostupom-skud/>.
8. 8. Биометрия. [Электроний ресурс]. – Режим доступа: <https://secur.ua/biometrija/>
9. 9. Петик В.А. Создание умного дома на базе Arduino / В.А. Петик // – М.: ДМК Пресс, - 2018. – 180 с.

10. 10. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things / В.А. Петин // - СПб.: БХВ-Петербург, 2016. -320 с.
11. 11. Геддес Марк. 25 крутых проектов с Arduino / Марк Геддес; [пер. с англ. М.А. Райтмана]. — Москва : Эксмо, 2019. — 272 с.
12. 12. Петин В. А. Проекты с использованием контроллера Arduino. - СПб.: БХВ-Петербург, 2014. — 400 с.
13. 13. Петин В. А. Практическая энциклопедия Arduino / В.А. Петик, А.А. Биняковский // – М.: ДМК Пресс, - 2017. – 152с.
14. 14. Соммер У. C61 Программирование микроконтроллерных плат Arduino/Freeduino / У. Соммер // - СПб.: БХВ - Петербург, 2012. - 256 с.
15. 15. Access control systems. [Электронный ресурс]. – Режим доступа: https://isbc.com/app_area/humans-id/access-control/.
16. 16. Система контроля и управления доступом. [Электронный ресурс]. – Режим доступа: https://esmart.ru/app_area/skud/skud--eto-sistema/.
17. 17. Среда разработки Arduino. [Электронный ресурс]. – Режим доступа: http://arduino.ru/Arduino_environment.
18. 18. Getting Started with Arduino UNO. [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/en/Guide/ArduinoUno>.
19. 19. Language Reference. [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/reference/en>.
20. 20. Arduino Create. [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/en/main/create>.
21. 21. DS75176B/DS75176BT Multipoint RS-485/RS-422 Transceivers – Режим доступа: <http://www.allcomponents.ru/pdf/nsc/ds75176.pdf>

22. 22. Podscheta. [Электроний ресурс]. – Режим доступа: <https://www.kkm.ru/blog/kak-vybrat-sistemu-podscheta-posetiteley-dlya-magazina/>.
23. 23. Schyotchik [Электроний ресурс]. – Режим доступа: <https://satu.kz/p71704199-schyotchik-posetitelej-analitikoj.html>
24. 24. Uchet [Электроний ресурс]. – Режим доступа: <https://www.uchet.biz/vidy-sistem-podscheta-posetitelei.php>
25. 25. Miniboard [Электроний ресурс]. – Режим доступа: <https://miniboard.com.ua/platy-rasshireniya/510-rs485-shield-dlya-arduino-dfrobot.html>
26. 26. Arduino [Электроний ресурс]. – Режим доступа: <https://doc.arduino.ua/ru/hardware/Leonardo>
27. 27. Arduino [Электроний ресурс]. – Режим доступа: <https://arduino-kit.com.ua/distancii-i-rasstoyaniya/>
28. 28. Arduino [Электроний ресурс]. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardUno>
29. 29. Miniboard [Электроний ресурс]. – Режим доступа: <https://miniboard.com.ua/platy-rasshireniya/510-rs485-shield-dlya-arduino-dfrobot.html>
30. 30. Arduino [Электроний ресурс]. – Режим доступа: <https://prom.ua/ua/p1535953071-plata-arduino-uno.html>

Додаток А

Зауваження нормоконтролера

Таблиця А.1 – Зауваження нормоконтролера

Сторінка	Позначення	Зміст зауваження
ПЗ		Посилання на джерела починаються з номеру 5, і не йдуть у порядку використання
Стр. 28-31, 40, 53-57		Невірні переліки
Перелік посилань		Подвійна нумерація

Додаток Б

Програма для мікропроцесорних плат Arduino Leonardo та Arduino UNO

Підпрограма візуалізації

// Робота з послідовним портом

import processing.serial.*;

Serial port;

// Множник розмірів кімнат для масштабування розмірів

int roomScale = 3;

// Координата початку малювання з X та Y

int drawStart = 75;

// ширина дверного отвару

int doorwayWidth = 15 * roomScale;

// масив ширин кімнат

int[] roomWidth = {

130 * roomScale, 60 * roomScale, 25 * roomScale, 85 * roomScale, 130 *
roomScale

};

// масив висот кімнат

int[] roomHeight = {

100 * roomScale, 90 * roomScale, 90 * roomScale, 85 * roomScale, 75 *
roomScale

};

// Кількість людей у кімнаті зберігається у цьому масиві

Integer[] persons = {

0, 0, 0, 0

};

// Поточний стан кожного отвору

```
// зберігається тут. 0 - без зміни, 1 - хтось вийшов, 2 - хтось увійшов
int[] doorwayState = {
    0, 0, 0
};
```

```
// Ця функція виконується один раз при старті скетчу
void setup()
{
    // розмір вікна
    size(800, 700);
    // створюємо послідовне з'єднання з Arduino на 17-му COM-порту
    port = new Serial(this, "COM17", 9600);
    // Дані з COM-порту буферизуються доти, доки не прийде '\n'
    port.bufferUntil('\n');
```

```
// Координати текстового поля задаються щодо центру тексту
textAlign(CENTER, CENTER);
// розмір тексту
textSize(50);
}
```

```
// Ця функція викликається щоразу, коли настав час малювати кадр
// Аналог loop() в Arduino
```

```
void draw()
{
    // Зафарбовуємо вікно чорним кольором
    background(0);
```

```
// Малювання кімнат
    drawRooms();
```

```
}
```

```
void drawRooms()
{
    // задаємо білий колір ліній
```

```
stroke(255);

// ширина ліній 10 px
strokeWeight(10);

// заливка об'єктів - сіра
fill(150);

// малювання кімнат
rect(drawStart, drawStart, roomWidth[0], roomHeight[0]);

rect(drawStart + roomWidth[0], drawStart, roomWidth[1], roomHeight[1]);

rect(drawStart + roomWidth[0] + roomWidth[1], drawStart, roomWidth[2],
roomHeight[2]);

rect(drawStart + roomWidth[0], drawStart + roomHeight[1],
roomWidth[3], roomHeight[3]);

rect(drawStart, drawStart + roomHeight[0], roomWidth[4],
roomHeight[4]);

// перевірка стану дверних отворів функції checkDoorway()
// якщо хтось пройшов
if (checkDoorway(0)) {
    // дверний отвір буде червоним
    stroke(255, 0, 0);
} else {
    // інакше - темно-сірий
    stroke(50);
}

// малювання отвору
line(drawStart + roomWidth[0],
drawStart + roomHeight[0] * 0.6,
```

```
drawStart + roomWidth[0],  
doorwayWidth + drawStart + roomHeight[0] * 0.6);
```

```
if (checkDoorway(1)) {  
    stroke(255, 0, 0);  
} else {  
    stroke(50);  
}
```

```
line(drawStart + roomWidth[0] + roomWidth[1],  
drawStart + roomHeight[0] * 0.15,  
drawStart + roomWidth[0] + roomWidth[1],  
doorwayWidth + drawStart + roomHeight[0] * 0.15);
```

```
if (checkDoorway(2)) {  
    stroke(255, 0, 0);  
} else {  
    stroke(50);  
}
```

```
line(drawStart + roomWidth[0] + roomWidth[1] + (roomWidth[2] -  
doorwayWidth)/2,  
drawStart + roomHeight[2],  
drawStart + roomWidth[0] + roomWidth[1] + doorwayWidth +  
(roomWidth[2] - doorwayWidth)/2,  
drawStart + roomHeight[2]);
```

```
// закінчення циклу
```

```
// Малювання дверних отворів, які не мають датчиків
```

```
stroke(50);
```

```
line(drawStart + roomWidth[0] + roomWidth[1] + roomWidth[2],  
drawStart + roomHeight[0] * 0.1,
```

```
drawStart + roomWidth[0] + roomWidth[1] + roomWidth[2],  
2 * doorwayWidth + drawStart + roomHeight[0] * 0.1);
```

```
line(drawStart + roomWidth[0],  
drawStart + roomHeight[0] + roomHeight[4] * 0.4,  
drawStart + roomWidth[0],  
doorwayWidth + drawStart + roomHeight[0] + roomHeight[4] * 0.4);
```

```
// У центрах кімнат вказується кількість осіб у кімнаті  
fill(255);
```

```
text(persons[0].toString(), drawStart + roomWidth[0]/2,  
drawStart + roomHeight[0]/2);  
text(persons[1].toString(), drawStart + roomWidth[0]+roomWidth[1]/2,  
drawStart + roomHeight[1]/2);
```

```
text(persons[2].toString(),  
drawStart + roomWidth[0] + roomWidth[1]+roomWidth[2]/2,  
drawStart + roomHeight[2]/2);
```

```
text(persons[3].toString(),  
drawStart + roomWidth[0]+roomWidth[3]/2,  
drawStart + roomHeight[1] + roomWidth[3]/2);
```

```
}
```

```
// Перевірка стану дверного отвору під номером currentDoorway  
boolean checkDoorway(int currentDoorway)  
{
```

```
// Якщо ніхто не проходив – повернемо false  
boolean result = false;
```

```
switch(doorwayState[currentDoorway]) {  
// Якщо хтось вийшов із кімнати  
case 1:
```

```

    result = true; // стан змінився
    // якщо у кімнаті було більше 0 осіб
    if (persons[currentDoorway] > 0) {
        // заберемо одну людину з цієї кімнати
        persons[currentDoorway]--;
    }
    // додавання його до сусідньої кімнати
    persons[currentDoorway+1]++;
    // зміна стану відпрацьовано
    doorwayState[currentDoorway] = 0;
    break;
// Якщо хтось увійшов до кімнати
case 2:
    result = true; // состояние изменилось
    // якщо у сусідній кімнаті більше 0 осіб
    if (persons[currentDoorway+1] > 0) {
        // заберемо одну людину із сусідньої кімнати
        persons[currentDoorway+1]--;
    }
    // і додамо до іншої кімнати
    persons[currentDoorway]++;
    // зміна стану відпрацьована, запам'ятаємо це
    doorwayState[currentDoorway] = 0;
    break;
}
// Повертаємо стан дверного отвору
return result;

}

// Ця функція працює при отриманні байта із послідовного порту
void serialEvent (Serial port)
{

    try {

```

```

// Читання даних від Arduino
String data = port.readStringUntil('\n');

// формат посилки x:y, де x-номер дверного отвору,
// y – його статус. Розбиваємо рядок data на кілька рядків,
// використовуючи ':' як роздільник
String[] list = split(data, ':');
// номер отвору міститься у першому рядку. Відразу переводимо
рядок у число
Integer doorwayNumber = int(list[0]);

// статус отвору - у другому рядку. Переклад рядок у число
// HARDCODE! substring(0, 1) - копіювання підрядки розміром 1
символ з рядка,
// початок з 0-го символу
Integer state = int(list[1].substring(0, 1));

// Зберігаємо стан дверного отвору
doorwayState[doorwayNumber] = state;

// запис у консоль, що надійшла інформацію для налагодження
println(data);
println(list[0]);
println(list[1]);

}

// Рішення проблеми
catch (Exception e) {
// нема зв'язку з Arduino
println("Connection...");
}
}

// Звільнення послідовного порту при закритті програми.
void stop() {

```

```
port.clear();  
port.stop();  
}
```

Провідне обладнання

```
#define STATE_BEGIN 0  
#define STATE_SENDED 1  
#define STATE_RECEIVED 2  
  
void setup()  
{  
    Serial.begin(9600);  
    Serial1.begin(1200);  
  
    while (!Serial) ;  
  
    pinMode(2, OUTPUT);  
}  
  
int curDev = 0;  
  
int state = STATE_BEGIN;  
  
void loop()  
{  
    static unsigned long t;  
    byte msg[4] = { 0x03, 0x20, 0x00, 0x05 };  
  
    if(state == STATE_BEGIN)  
    {  
        msg[1] = 0x20 + curDev;  
  
        digitalWrite(2, HIGH);  
        Serial1.write((const byte*)msg, 4);  
        Serial1.flush();  
    }  
}
```



```

digitalWrite(2, LOW);

state = STATE_SENDED;
t = millis();
}
else if(state == STATE_SENDED)
{
    if(Serial1.available())
    {
        static byte imsg[4];

        imsg[0] = imsg[1];
        imsg[1] = imsg[2];
        imsg[2] = imsg[3];
        imsg[3] = Serial1.read();

        if(imsg[0] == 0x03 && imsg[3] == 0x05)
        {
            Serial.print(curDev);
            Serial.print(':');
            Serial.println((int)(imsg[1]));
            Serial.flush();
            delay(1);

            state = STATE_RECEIVED;
        }
    }
    else if(millis()-t > 50) {
        state = STATE_RECEIVED;
    }
}
else if(state == STATE_RECEIVED)
{
    if(++curDev == 15)
    {
        curDev = 0;
    }
}

```

```

    }

    state = STATE_BEGIN;
}
}

```

Підпрограма ведені пристрої

```

signed long t1 = 0, t2 = 0;
unsigned char dir = 0;

```

```

#define BASE_ADDR 0x20
#define MSG_LEN 4

```

```

void sensor1()
{
    t1 = millis();

    check_motion();
}

```

```

void sensor2()
{
    t2 = millis();

    check_motion();
}

```

```

void check_motion()
{
    if(t1 == 0 || t2 == 0) return;

    if(t2-t1 > 0)
    {
        tone(9, 100, 1000);
        dir = 1;
    }
}

```

```

    }
    else if(t1-t2 > 0)
    {
        tone(9, 300, 1000);
        dir = 2;
    }

    t1 = t2 = 0;
}

void dispatch_msg(byte* msg)
{
    if(msg[1] != BASE_ADDR+addr()) return;

    msg[0] = 0x03;
    msg[1] = dir;
    msg[2] = 0x00;
    msg[3] = 0x05;

    digitalWrite(2, HIGH);

    Serial.write(msg, 4);
    Serial.flush();

    digitalWrite(2, LOW);

    dir = 0;
}

void setup(void)
{
    Serial.begin(1200);

    digitalWrite(13, LOW);
    digitalWrite(4, HIGH);

```

```
pinMode(8, INPUT);  
pinMode(9, INPUT);  
pinMode(10, INPUT);
```

```
pinMode(2, OUTPUT);  
digitalWrite(2, LOW);
```

```
    analogReference(INTERNAL);  
}
```

```
unsigned char addr(void)  
{  
    unsigned char a = 0x00;  
  
    if(digitalRead(8)) a |= 0x01;  
    if(digitalRead(9)) a |= 0x02;  
    if(digitalRead(10)) a |= 0x04;  
  
    return a;  
}
```

```
byte msg[4];
```

```
void loop(void)  
{  
    static unsigned long n1 = 0, n2 = 0;  
  
    if(analogRead(A0) > 500)  
    {  
        if(++n1 >= 20)  
        {  
            if(n1 > 20+1) n1--;  
  
            if(n1 == 20) sensor1();  
        }  
    }  
}
```

```
else n1 = 0;
```

```
if(analogRead(A1) > 500)
```

```
{
```

```
    if(++n2 >= 20)
```

```
    {
```

```
        if(n2 > 20+1) n2--;
```

```
        if(n2 == 20) sensor2();
```

```
    }
```

```
}
```

```
else n2 = 0;
```

```
if(Serial.available())
```

```
{
```

```
    int i;
```

```
    byte b;
```

```
    for(i = 0; i < MSG_LEN-1; i++)
```

```
        msg[i] = msg[i+1];
```

```
    msg[MSG_LEN-1] = Serial.read();
```

```
    if(msg[0] == 0x03 && msg[3] == 0x05)
```

```
    {
```

```
        dispatch_msg(msg);
```

```
    }
```

```
}
```

```
}
```

Додаток В
Перелік копій демонстраційного матеріалу

Слайд 1 – Тема дослідження

Слайд 2 – Актуальність теми дослідження

Слайд 3 – Призначення системи підрахунку людей

Слайд 4 – Типи технологій системи підрахунку відвідувачів

Слайд 5 – Принцип дії лічильників на основі інфрачервоних променів

Слайд 6 – Розміщення обладнання ІЧ вертикальних лічильників в отворі

Слайд 7 – Епюри напруги лічильників при проходженні людини через отвір

Слайд 8 – Обладнання для розробки мережі лічильників

Слайд 9 – Основні відомості для розробки мережі лічильників

Слайд 10 – Блок схеми програмного забезпечення системи підрахунку кількості людей в приміщеннях офісу

Слайд 11 – Схема мережі системи підрахунку людей в кімнатах офісу

Слайд 12 – Висновки