Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Оцінка ризиків кіберфізичних систем на базі мікроконтролерів типу Arduino

Виконала: студентка 6 курсу, групи СБм-61
спеціальності 125 - Кібербезпека
(шифр і назва спеціальності)

_____ Ярощук І. В.
(підпис) (прізвище та ініціали)

Керівник _____ Скоренький Ю. Л.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Лобур Т. Б.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Загородна Н. В.
(підпис) (прізвище та ініціали)

Рецензент _____ Баран І. О.
(підпис) (прізвище та ініціали)

Тернопіль
2020

АНОТАЦІЯ

Кваліфікаційна робота магістра на тему: «Оцінка ризиків кіберфізичних систем на базі мікроконтролерів типу Arduino» Ярощук Інни Віталіївни – Тернопільський національних технічний університет, Факультет комп`ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2020. С. – 98, рис. – 11, табл. – 18, додатки – 9.

Метою кваліфікаційної роботи є удосконалення технології оцінки ризиків інформаційної безпеки КФС та розроблення рекомендацій щодо оцінки ризиків при прототипуванні КФС та мінімізації ризиків при експлуатації КФС.

В процесі дослідження використано загальнонаукові методи пізнання: порівняння, системний аналіз, моделювання. Також були проведені експериментальні вимірювання та здійснено математичне опрацювання з метою отримання кількісної оцінки стану інформаційної безпеки.

На основі проведеного порівняльного аналізу з врахуванням специфіки кіберфізичних систем обрано методи оцінки ризиків CRAMM та FoMRA. Створено прототип на платформі Arduino та проведено його тестування щодо придатності для забезпечення захисту і безпечного функціонування інформаційно-телекомунікаційних систем. Для виявлених ризиків було розроблено контрзаходи, при виконанні яких система може вважатися безпечною.

Ключові слова: КІБЕРФІЗИЧНА СИСТЕМА, ІНФОРМАЦІЙНА БЕЗПЕКА, ОЦІНКА РИЗИКІВ, КОНТРОЛЬ ДОСТУПУ.

ABSTRACT

Qualification work for the master degree on the topic: «Risk assessment of cyberphysical systems based on Arduino microcontrollers» by Yaroshchuk Inna - Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group SBm-61 // Ternopil. Pages - 98, figs. - 11, tables - 18, appendices - 9.

The purpose of the qualification work is to improve the technology of risk assessment for information security of CPS and to develop recommendations for risk assessment in prototyping of CPS and minimization of risks during CPS use. During the research, general scientific methods of comparison, system analysis, modeling were used. Experimental measurements were also performed and mathematical processing was done in order to obtain a quantitative assessment of information security. Based on the conducted comparative analysis, taking into account the peculiarities of cyberphysical systems, the methods of CRAMM and FoMRA risk assessment were chosen. A prototype was created on the Arduino platform and tested to ensure the protection and safe operation of information and telecommunications systems. Countermeasures have been proposed for the identified risks to ensure information safety of the system.

Keywords: CYBERPHYSICAL SYSTEM, INFORMATION SECURITY, RISK ASSESSMENT, ACCESS CONTROL

СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

КФС – Кіберфізична система;

НС – Надзвичайна ситуація;

НСД – Несанкціонований доступ;

ОС – Операційна система;

ПЗ – Програмне забезпечення;

РЕА - Радіо-електронна апаратура;

CRAMM – (англ. CCTA Risk Analysis & Management Method) метод CCTA аналізу та контролю ризиків;

DoS – (англ. Denial of Service) відмова в обслуговуванні;

FMEA – (англ. Failure Mode and Effects Analysis)метод аналізу відмов та наслідків;

FoMRA – (англ. formal model of risk assessment) формалізований метод оцінки ризиків;

GTST-MLD - (англ. goal tree success tree master logic diagram) метод дерев успіху, дерев цілей та головної логічної діаграми;

HAZOP - (англ. Hazard and Operability Study) методологія дослідження небезпеки та працездатності;

NIST- (англ. National Institute of Standards and Technology) Національний інститут стандартів і технологій США;

STAMP - (англ. System-Theoretic Accident Model and Processes) системно-теоретична модель збоїв та процесів;

RFID - (англ. Radio frequency identification) — радіочастотна ідентифікація;

USB – (англ. Universal Service Bus) універсальна шина обміну даними.

ЗМІСТ

ВСТУП

Актуальність теми. Інформаційна безпека є важливою складовою інформаційного технологій, які використовуються сучасним суспільством та повинна гарантувати цілісність, конфіденційність та доступність інформаційних ресурсів. Проектування та експлуатація інформаційних систем повинні включати аналіз і управління інформаційними ризиками. Для оцінки інформаційного ризику необхідно якісно чи кількісно виміряти його рівень та співставити його з певним гранично допустимим значенням. При цьому визначаються як ймовірність настання події, так і розміри її можливих наслідків.

Наукова новизна роботи:

- проведено порівняльний аналіз методологій FMEA, CRAMM, HAZOP, FoMRA та обґрунтовано вибір методу для оцінювання ризиків інформаційної безпеки з урахуванням особливостей функціонування КФС.
- проведено експериментальне дослідження компонентів КФС для оцінки ризиків для інформаційних систем, побудованих із застосуванням КФС.

Практичне значення дослідження полягає у розроблених рекомендаціях щодо оцінки ризиків при прототипуванні КФС та мінімізації ризиків при експлуатації КФС.

Метою дипломної роботи є удосконалення технології оцінки ризиків інформаційної безпеки КФС.

Реалізація мети дослідження вимагала розв’язання таких завдань:

- проаналізувати стан дослідженості проблеми в літературних та інтернет-джерелах;
- провести порівняння методів оцінки ризиків інформаційної безпеки КФС та обґрунтувати критерії вибору цього методу для практичного застосування;

- провести оцінку рівня захищеності ресурсів в інформаційних системах на базі КФС;
- оцінити можливість реалізації потенційних загроз інформації та можливість несанкціонованого доступу до елементів інформаційної системи на основі КФС.

Об’єктом дослідження дипломної роботи є ризики інформаційної безпеки КФС.

Предметом дослідження є оцінювання ризиків інформаційної безпеки компонентної бази КФС та КФС вцілому з використанням апробованих методів.

Методи дослідження. В процесі дослідження використано загальнонаукові методи пізнання: порівняння, системний аналіз, моделювання. Також були проведені експериментальні вимірювання та здійснено математичне опрацювання з метою отримання кількісної оцінки стану інформаційної безпеки.

Апробація результатів дослідження. Основні положення та результати дослідження обговорювалися та були схвалені на ІІІ міжнародній студентській науково-технічній конференції «Природничі та гуманітарні науки. Актуальні питання» та на VIII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 2020р.).

Структура роботи. Дипломна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел із 50 найменувань, дев’яти додатків. Робота містить 11 рисунків, 18 таблиць і 12 формул. Обсяг основного тексту становить 55 сторінок, перелік використаних джерел - 6 сторінок, додатки - 28 сторінок. Загальний обсяг кваліфікаційної роботи складає 98 сторінок.

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Загальна характеристика та призначення КФС

Термін кіберфізичні системи означує певну сукупність обчислювальних та фізичних складових, що спроєктовані на взаємодію між собою у вигляді єдиної системи яка здатна адаптовуватись під зміни фізичного середовища [1]. У праці [2] вказано, що кіберфізичні системи - це керовані, надійні та розширювані мережеві фізичні системи, які в подальшому інтегровані з обчислювальними, комунікаційними та контрольними можливостями, які можуть взаємодіяти з людьми за допомогою багатьох нових способів. По суті можна сказати, що КФС - нове покоління систем із інтегрованими обчислювальними та фізичними можливостями, які виникли завдяки об'єднанню обчислень, зв'язку та управління.

КФС стали основою та ядром «Промисловості 4.0» та «Промислового Інтернету». Виходячи з вищенаведеної інформації випливає, що КФС – це системи Інтернету речей, які використовують властивості контролю та моніторингу в реальному часі. КФС можна по суті розглядати як фізичний процес і відповідні цьому процесу системи керування, що підключені завдяки певним формам загальних мереж зв'язку [3], як показано на рисунку 1.1.

Рисунок 1.1 – Схематика КФС

На сьогодні КФС широко поширені по всьому світу. NIST було розроблено класифікацію кіберфізичних систем [4], до якої входять системи розумного автономного виробництва, розумних енергосистем [5], розумних будинків, розумних конструкцій, розумного транспорту та розумні охорона здоров'я та безпека життя.

Кіберфізичні системи можна описати як тріаду, яка являє собою взаємодіючі технології датчиків, вбудованої системи та мережевих технологій. Цілісна КФС залежно від потреб користувача повинна містити такі характеристики: надійність, паралельність, робота в реальному часі, обробка великих масивів інформації, поширеність, автономність, безпека, динамічна реорганізація та неоднорідність.

Надійність. КФС працюють в середовищі, де весь час щось змінюється. Отже важливою властивістю такої системи буде здатність самостійно справлятися з несподіваними ситуаціями та несправностями підсистеми. Виконання цієї властивості як умови буде гарантувати безпечність використання КФС людиною без шкоди для її життя та здоров'я.

Паралельність. Основною особливістю події в фізичному середовищі є паралельність [6]. Паралельність це можливість одночасно реалізовувати фізичні та обчислювальні процеси для повноцінної роботи КФС.

Робота в реальному часі. Основою роботи інформаційного світу в режимі реального часу базується на тому, що в інформаційному світі подія відображається в той же самий час, що і в фізичному світі. Щоб система могла сприймати фізичний процес та мала змогу втручатися у фізичні процеси у режимі реального часу, КФС мусить підтримувати постійну взаємодію між фізичними і обчислювальними процесами, що є дуже високою вимогою для продуктивності.

Обробка великих масивів інформації. Через те, що КФС це сукупність фізичної, обчислювальної та мережевої взаємодії, обсяг інформації що отримується та обробляється системою дуже великий. Як приклад це може бути інформація отримана з таких вузлів як температура, напруга, струм, та ін. Кількість таких вузлів може бути критично велика, що робить цю властивість загальною для всіх КФС.

Поширеність. КФС – типова розподілена обчислювальна мережа [6], що містить у своєму складі багато вбудованих систем з обмеженою ємністю кожного вузла.

Автономність. Завдяки властивості автономності, система має здатність самостійно реагувати на зміни в фізичному світі, і цим забезпечувати нормальну роботу КФС.

Безпека. Оскільки КФС працює з фізичним середовищем, обов'язковою умовою є забезпечення високого рівня безпеки.

Динамічна реорганізація. У зв'язку зі швидкою зміною процесів фізичного середовища КФС потрібно навчити самостійно налаштовуватись під кожну зміну, реконфігуруватися. У протилежному випадку система може стати нестабільною або ж небезпечною для життя і здоров'я користувачів.

Неоднорідність. Через велику множину компонентів для розробки КФС (датчики, пам'ять, процесори, плати та ін.), що зараз виробляють, можна обрати будь-який елемент під власні потреби. Немає чітких правил, для вибору виробника того чи іншого елемента, тому потрібно зауважити, що всі елементи системи повинні бути налаштовані на взаємодію між собою та КФС.

1.2 Архітектура КФС

Головною властивістю КФС є можливість сприйняття навколишнього середовища. За допомогою циклу зворотного зв'язку інтерактивних обчислювальних процесів та фізичних процесів в КФС реалізується глибока інтеграція та взаємодія, розширення нових функцій, та з'являється можливість виявлення та контролю фізичних сутностей в реальному часі.

У праці [7] Тан описав архітектуру кіберфізичних систем. Основою для них є фізичний світ. У фізичному світі знаходиться джерело інформації всієї КФС, та відбувається остаточний зворотній зв'язок. Фізичний рівень можна поділити на дві частини: фронтенд (інтерфейс) та бекенд (програмно-апаратна частина). Фронтенд фізичного рівня КФС також можна поділити на дві частини: частина виконання, та частина сприйняття. КФС у фізичному середовищі оперують всіма

видами фізичних вузлів, що можуть знаходитись дуже близько, або дуже далеко один від одного. Тому існує деяка проблема, підтримки взаємодії між собою цих вузлів. Цю проблему вирішує мережа, вона закладає собою основу для обміну інформації. Побудова мереж КФС є набагато складнішою за побудову звичайної комп'ютерної мережі, тому що кіберфізичні мережі потребують не лише створення основи для застосування мережі спільного використання ресурсів, але і існує потреба захисту неоднорідності великих фізичних вузлів, та реалізації безперебійних з'єднань системи. Враховуючи технології маршрутизації, контролю доступу та передачі, для КФС важлива наявність технологій підтримки, таких як наприклад технології семантичного аналізу та опису неодорідних даних, які генерують гетерогенні вузли , технології перевантаження мережі для вирішення масової передачі даних, технології зберігання масивних даних, технології локалізації вузлів [6].

Ще одна важлива частина КФС – блок управління КФС. Датчики отримують інформацію про події фізичного світу, та передають її на блок управління, який обробляє її, та генерує оброблену інформацію з урахуванням певних правил. Далі відбувається передача згенерованої інформації каналами мережі КФС до модуля реалізації та блоку сканування, після чого завершить інтерактивний процес всієї системи.

Додатково до архітектури кіберфізичних систем також є доцільним розділення КФС на модель ієрархічної структури. Залежно від функцій комплектуючих частин, КФС можна розділити на фізичний, мережевий та прикладний рівень.

Фізичним рівнь це взаємодія фізичних пристроїв з навколишнім середовищем. Це ієрархія системних вузлів, де у фізичному процесі вбудована деяка обчислювальна функція. На цьому рівні головними є датчики, що сканують навколишнє середовище, виконавчі пристрої, які реагують на стан навколишнього середовища, енергоблоки, які передають енергію на інші прилади, та ін [6].

В ієрархічній структурі на мережевому рівні основною функцією є здійснення мережевої передачі. На мережевому рівні відбувається забезпечення

підтримки взаємозв’язку між стільниковими елементами КФС та безпеки неоднорідності одиниць фізичного рівня. У роботі [8] автори запропонували концепцію Кіберфізичного інтернету, для отримання мережі фізичних обчислень, що є доволі подібною до мережі Інтернет, з метою підтримання взаємозв’язку та взаємодії між всіма пристроями КФС. Це покращує можливості передачі в мережі, вирішує проблеми навантаження, що відбувається через передачу великого обсягу даних через мережу.

Прикладний рівень є відповідальним за всю КФС представлену користувачеві. На цьому рівні відбувається інкапсуляція даних фізичного та мережевого рівнів, і проводиться генерація різних модулів програми. Прикладний рівень в основному збирає вимоги до завдання [6], та розумно ставить завдання, після чого відповідно до заданих завдань налаштовує ресурси, позиціонує та планує ресурси для виконання конкретних завдань.

1.3 Атаки на КФС

У зв’язку зі зростанням кількості кібератак по всьому світу, безпека та конфіденційність на сьогодні є одними з найбільших проблем [9] для користувачів та різних компаній. Важливо забезпечити конфіденційність інформації, що передається відкритими мережами між джерелом і пунктом призначення. Є велика вірогідність проведення кібератаки на цю мережу чи системні пристрої, щоб порушити цілісність інформації чи використання пристроїв для особистого зловживання.

Для моніторингу та керування, передача даних між фізичними процесами та системою керування відбувається за допомогою комунікаційних мереж [10, 11]. Оскільки такі мережі аналогічно можуть використовувати потенційні порушники, це дає їм змогу отримати доступ до потенційних точок входу, для втручання в передачу даних. Використавши слабкі місця в мережах чи протоколах зв’язку, порушники можуть отримати доступ до критичної інформації про систему. Вони можуть маніпулювати переданими даними для порушення фізичного процесу за допомогою кіберзасобів, які згадуються в праці

[12] як кіберфізичні атаки. Якщо такі атаки спрямувати на процеси які є критичними для безпеки КФС, буде завдано величезну шкоду, як фізичним елементам системи, так і можливо загрожувати життю людей.

Слабкою стороною КФС, яка дозволяє зловмисникам зменшити рівень надійності є вразливості. Вони виникають в наслідок слабкої політики безпеки, дизайну системи, чи проблем програмно-апаратного забезпечення КФС [13] Найпоширенішими вразливості - це вразливості програмного забезпечення, апаратні, технічні, вразливості мережі, управління та платформи [14, 15].

Апаратна вразливість безпосередньо стосується апаратної частини КФС, яка недостатньо захищена. Таку вразливість важко виявити та дуже важко виправити. Запобігти цій вразливості дуже важко, але зменшити її вплив можна за допомогою обмеження фізичного доступу до обладнання та приміщення де воно розміщене.

Вразливості програмного забезпечення можна виявити в операційних системах, прикладному програмному забезпеченні, у протоколах зв’язку, і помилках в роботі КФС. Такі вразливості можуть бути використані порушником для саботування системи, зміни та модифікації нормального стану і запуску системи. Інша поширена вразливість програмного забезпечення є відсутність шифрування даних, введення команд ОС, введення SQL, переповнення буфера, відсутність автентифікації для критичної функції, відсутність авторизації, необмежене завантаження небезпечних типів файлів, довіра до невірних входів у рішенні безпеки, міжсайтовий сценарій та підробка, Завантаження кодів без перевірки цілісності, Використання непрацездатних алгоритмів, перенаправлення URL-адрес на ненадійні сайти, обхід шляху, помилки, слабкі паролі та програмне забезпечення, яке вже заражене вірусом.

Технічні вразливості найчастіше виникають через помилки здійснені самими користувачами системи.

Через недосконалу розробленість або відсутність політики безпеки виникають вразливості управління.

Вразливості платформи крім відсутності заходів захисту, включають в себе вразливості конфігурації програмного та апаратного забезпечення

Вразливості мережі являють собою вразливості та недоліки конфігурації та мережевого обладнання.

Існує безліч класифікацій атак на КФС, але найпоширенішими є кіберзагрози, шкідливе програмне забезпечення, криптографічні та мережеві атаки.

Шкідливе програмне забезпечення використовується для обходу засобів контролю доступу до системи, компрометації функцій КФС та викрадення або знищення інформації. Заподіяння шкоди хост-комп'ютеру є основною метою зловмисного ПЗ [16].

Зловмисні атаки називають кіберзагрозами. Для порушення цілісності організації чи персональних систем, ці атаки знаходять слабкі місця КФС. Метою кіберзагрози є пошкодження або відключення роботи системи [17]. Чжан та інші у своїй праці [17] назвали багато типів кіберзагроз, і вони можуть походити з первинних джерел: природи (землетруси, урагани, повені та пожежа) та людей, фізичних атак, несправності обладнання, несправності лінії, електромагнітних витоків та електромагнітні перешкоди.

Процедура уникнення безпеки системи шляхом виявлення слабких місць шифру, алгоритму безпеки, шаблонів управління ключами, операційних системах та криптографічних протоколах називається криптографічною атакою, також її ще називають криптоаналізом [18].

Мережеві атаки бувають двох видів: активні і пасивні. Під час активної атаки відбувається зміна даних, а під час пасивної відбувається лише контроль та моніторинг даних, що проходять через мережу. Поширені мережеві атаки - це прослуховування мережі, модифікація даних, підробка особистих даних, атаки на основі паролів, відмова в обслуговуванні, людина посеред, атака на скомпрометовані ключі, sniffer, атака рівня додатку, атаки доступу, розвідувальні атаки, напади на приватне життя та руйнівні атаки [13].

Внутрішні, або ж як їх ще називають інсайдерські атаки на КФС зручно класифікувати за видами, точками входу та прихованими умовами [19]. Порушники мають змогу підробляти пакети даних, які отримує контролер, переривати комунікаційні мережі або підслуховувати мережу [20]. Виходячи з

сказаного вище можна відповідно класифікувати атаки на три групи: обманні атаки, відмова в обслуговуванні та підслуховування [3]. У таблиці 1.1 наведено опис цих груп атак.

Таблиця 1.1- Найвідоміші типи атак [19] на КФС

Тип атаки		Опис атаки
Підслуховування (Eavesdropping)		Скомпрометуйте систему та підслуховуйте передані дані
Відмова в обслуговуванні (Denial of Service (DoS))		Заглушіть мережевий трафік, щоб зробити канали зв'язку недоступними
Обманні атаки (Deception attack)	General deception attack	Перервіть передачу даних та введіть шкідливі дії
	False-data injection attack	Зміна переданих даних прихованим способом
	Replay attack	Використовуйте дані історії, щоб приховати поточну шкідливу дію
	Covert attack	Координуйте сигнали управління та вимірювання датчиків, щоб приховати дію атаки

Обманні атаки мають здатність порушувати цілісність пакетів даних, підробляючи передані дані між фізичним та кіберрівнем. Атаки на відмову в обслуговуванні за допомогою переривання комунікаційних мереж переривають передачу даних. Атака підслуховування спрямована на перехоплення мережевого трафіку і отримання відповідної інформації для подальшого аналізу, але така атака є пасивною і не впливає на фізичний процес.

1.4 КФС на платформі Arduino

Для розробки КФС існує велика множина програмного і апаратного забезпечення. Безліч різних датчиків для отримання даних про стан фізичного середовища можна використовувати не тільки в межах одного виробника. Датчики вимірюють фізичні властивості середовища та фізичної особи [21]. Варіантом такого симбіозу може бути використання плат на основі мікроконтролерів Arduino.

На сьогоднішній день Arduino можна вважати однією з найпопулярніших фізично-програмних платформ [22]. Arduino – програмно-апаратна платформа з відкритим кодом. Це друкована плата з невеликим електронним пристроєм, що дозволяє маніпулювати великими множинами різнотипних датчиків, електродвигунами, освітленням, дозволяє проводити передачу та отримання інформації каналами зв'язку тощо. Це симбіоз двох компонентів: електронний блок та програмне забезпечення.

Електронний блок Arduino являє собою плату з процесором типу ATmega328 та флеш пам'яттю 32 Кб. Також плата містить 14 цифрових контактів, які можуть використовуватись як вхід або вихід, і 6 аналогових входів [23]. Також допускається використання шести цифрових входів для забезпечення аналогового входу з широтно-імпульсною модуляцією. На платі можна використовувати різні протоколи зв'язку. Також кожна плата містить кнопку скидання та ланцюг послідовного програмування, як стандартні функції. Живлення плати можна організувати за допомогою USB, так і за допомогою постійного струму. Існує багато модифікацій даних плат, наприклад Arduino Nano, Uno, Leonardo, Mega, Duemilanove та ін. Для розробки більш ресурсозатратних проектів можна використовувати Arduino Mega1280 із 128 КБ пам'яті або новішу Arduino Mega2560 із 256 КБ пам'яті, вони мають більший об'єм пам'яті та більшу кількість виводів.

Щоб розробити КФС, однієї плати і датчиків мало. Для того, щоб створити систему, потрібно задати процесору певні команди, які в свою чергу будуть виконувати функції КФС, для цього потрібне середовище програмування. Найкращим, чи найзручнішим середовищем програмування є програма Arduino IDE. Arduino IDE має в собі всі необхідні функції для програмування Arduino, а також містить рід прикладних програм, що демонструють як підключати та взаємодіяти з різними пристроями, такими як датчики, дисплеї, світлодіоди, сервоприводи та ін. Для створення скетчів програми використовується С-подібна мова програмування спеціально розроблена для контролерів Arduino. ПЗ Arduino є безкоштовним і доступним для Windows, Mac OS X та Linux.

1.5 Методи оцінки ризику безпеки для КФС

Як сказано в документі [24], система оцінки та керування ризиками повинна охоплювати три складових: кібербезпеку, фізичну безпеку та функціональну безпеку. Фізична безпека являє собою попередження та захист від стихійних лих, вибухів, пожеж, потенційних аварій транспортних засобів, викидів хімічних речовин, радіаційне забруднення та ін. Функціональна безпека – це забезпечення доступності пристроїв, а також відмов задля підтримання безпечного стану процесу [25]. Кібербезпека стосується кіберсередовища організації або ж її уповноважених користувачів, що включає в себе всі пристрої та програмне забезпечення, процеси та інформацію яка фігурує в системі тощо.

Для розробки КФС оцінка та керування ризиками є пріоритетним завданням [26]. На початкових етапах розробки КФС потрібно передбачити всі можливі ризики які можуть бути спричинені різними чинниками, середовищем, а також ризики зв’язку середовища та КФС, ризики спричинені самою КФС, а також ризики між КФС і користувачами. У стандарті IEC61508 [27] визначено граничне значення безпечної несправності і безпечний рівень цілісності, що підтверджує ступінь безпечності системи. Даний стандарт визначає ризик-орієнтований підхід для забезпечення виконання функцій безпеки систем.

Система оцінки та управління ризиками зосереджує свої можливості на виявленні та аналізі вразливих місць КФС та оцінці збитків, які можуть бути завдані [1]. Оцінка ризиків поділяється на два види, - це якісна та кількісна оцінка. Перший вид залежить від досвіду експертів, а другий – дає точне обчислення ризику системи [1]. На сьогодні існує доволі велика кількість добре розроблених методологій для оцінки ризику безпечної розробки та роботи КФС. У цьому розділі буде наведено кілька основних типових технологій, які аналізують стан безпеки КФС.

Функціональною основою надійності та аналізу ризиків КФС можна назвати метод дерев успіху, цілей та головну логічну діаграму[28]. На рисунку 1.2 зображена комбінована модель GTST-MLD. У GTST-MLD функції та цілі – властивості, тоді як об’єкти та зв’язки між ними - дерева успіху та головна

логічна діаграма, побудовані за допомогою булевої логіки. З точки зору нечітких, логічних та фізичних взаємозв'язків модель може представляти складні фізичні системи. Три-ступенева модель для аналізу надійності представлена Брісco та ін [29] є розширеною моделлю GTST-MLD, що інтегрує несправності та збої.

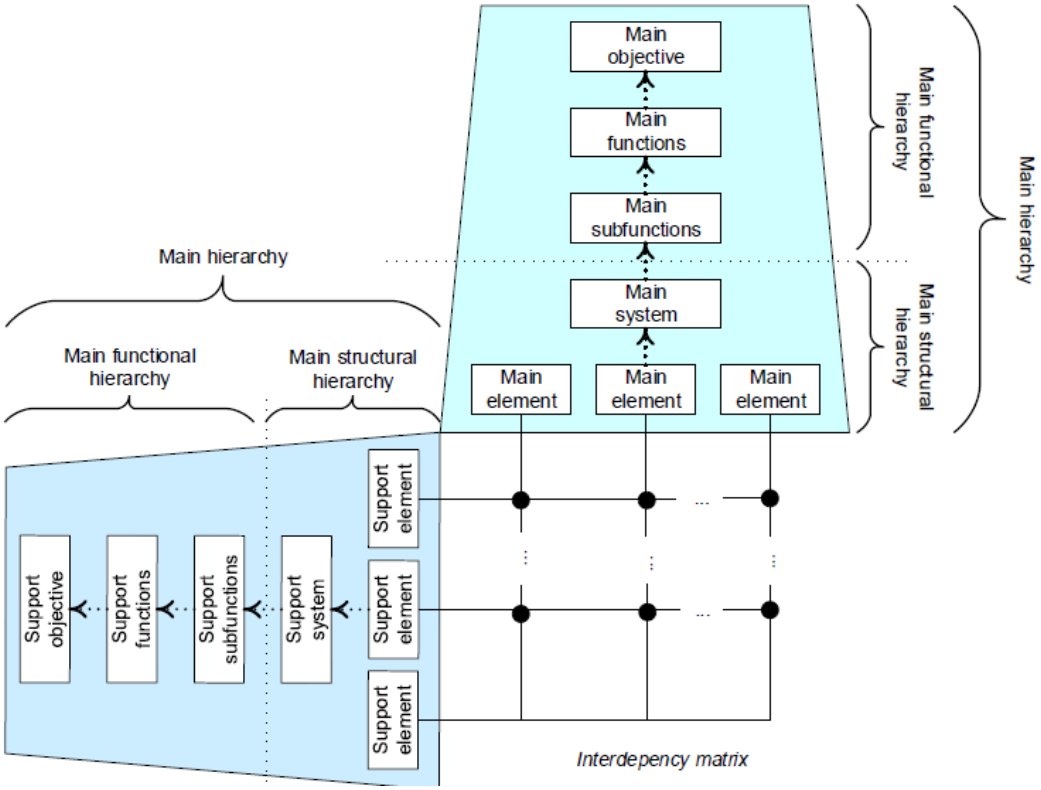


Рисунок 1.2 – Модель GTST-MLD [29]

Методологя небезпеки та працездатності (HAZOP) являє собою методику аналізу небезпеки процесів, що використовується для вивчення проблем працездатності і небезпек системи шляхом вивчення наслідків будь-яких відхилень від спроєктованих умов [30]. HAZOP за допомогою виявлення можливих небезпек та потенційних помилок експлуатації може визначити, яким способом відбувається відхилення процесу від спроєктованих умов, та побачити процес переходу в помилку чи несправність [31].

Однією з нових методик оцінки ризику є системно-теоретична модель збоїв та процесів (STAMP) яку розробив Левесон [32]. Під час виконання STAMP система визначається як ієрархічна структура управління. Для кожної взаємодії між шарами структури управління вводяться обмеження на поведінку

компонентів на нижчих шарах. Накладання таких обмежень на компоненти, дає можливість впливу на поведінку системи. На кожному шарі структури управління відбувається операції, що базуються на роботі циклу управління зворотнім зв'язком. Метою методу системного теоретичного аналізу процесів є можливість визначення всього сценарію процесу збою, і збоїв окремих компонентів системи. Даний спосіб дозволяє визначити причину виникнення небезпек.

На ранніх етапах проектування життєвого циклу КФС використовують метод аналізу відмов та аналізу наслідків (FMEA). В роботі [33] автор говорить що аналіз режимів відмов та аналіз наслідків – це структурований метод, заснований на команді, який проводить аналіз безпеки системи, та проводить оцінку потенційних збоїв і їх можливих наслідків. Для оцінки ступеня тяжкості відмов різних режимів проводять аналіз ефектів. На рисунку 1.3 зображено етапи роботи FMEA. Нуріан та ін. у [34] стверджує, що номер пріоритетності ризику є результатом вираженості, імовірності виникнення та імовірності виявлення, та є частиною кількісного аналізу методу аналізу відмов та аналізу наслідків.

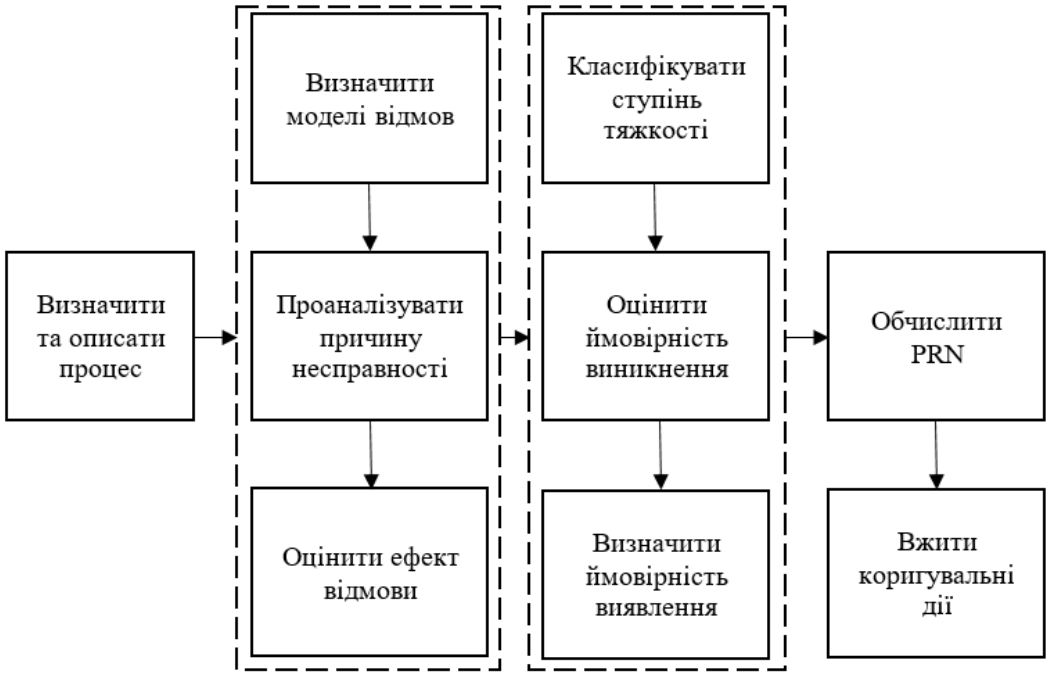


Рисунок 1.3 – Етапи FMEA

Метод оцінки ризиків CRAMM [35] проводить ідентифікацію та обчислення рівнів вразливостей на основі оцінювання, яке присвоюється для ресурсів, вразливостей ресурсів, та загроз. За допомогою ідентифікації та розроблення контрзаходів відбувається контроль ризиків, завдяки чому можна знизити їх рівень до допустимого. За допомогою цього методу можна переконатися, що система повністю захищена та всі проведені заходи були ефективними.

Побудований на основі методів CRAMM та MEHARI формалізований метод оцінки ризиків FoMRA [36] дозволяє оцінити всі можливі ризики за допомогою даних експертної оцінки та формальної математичної моделі визначення величини міри ризику. Даний метод, які і метод CRAMM є універсальний, тому підходить як для бізнес процесів, так і для КФС.

Для проведення оцінки ризику проєктовуваної КФС було обрано два найбільш зручних методи оцінки ризику: CRAMM та FoMRA.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Підходи до рівнів організації КФС

Для забезпечення правильного функціонування та безпеки КФС, при розробці потрібно враховувати всі можливі аспекти і технологічні підходи. У роботі [21] автор запропонував використовувати для розробки КФС підходи до рівнів пристроїв, комунікації і службового рівня.

Рівень пристроїв. На гарантіях, що підтримує пристрій зосереджено всю безпеку рівня пристроїв. Отож можна говорити, що пристрою типу розумних КФС, які використовують як домашні системи IoT потребують багато ресурсів, оскільки вони працюють майже завжди цілодобово і без потреби вимкнення. Плати що використовуються в таких системах зазвичай використовують слабкі процесори з малою пропускною спроможністю і низькою тактовою частотою. Такі критерії роблять невигідним використання надійних криптоалгоритмів. Також можна сказати, що така система повинна мати велику кількість оперативної пам’яті та на накопичувачах. Оскільки забезпечення ресурсами не єдиною проблемою розробки КФС, потрібно забезпечувати безпеку передачі даних дротовими і бездротовими каналами зв’язку. Багато КФС мають вразливість бути доступними для фізичного втручання. Дана вразливість дозволяє провести атаку за допомогою впливу на апаратні компоненти системи і втрутитися до роботи КФС, що є порушенням цілісності та конфіденційності системи. Отже пакети які передаються в мережі зв’язку КФС потребують захисту від НСД. Також потрібно забезпечити безпеку роботи системи віддаленого керування КФС, оскільки більшість КФС керуються віддалено, наприклад за допомогою телефона чи спеціального пульта, або взагалі голосових команд.

Рівень комунікації. Потрібно зауважити, що КФС потребують використання спеціальних динамічних протоколів та приватних віртуальних мереж задля ефективної роботи системи зв’язку. Доцільно зауважити, що використання міжмережевих екранів, шлюзів і хмарних технологій задля

можливості реалізації даного підходу. Також потрібно усунути проблему локальних атак та зараження шкідливим програмним кодом, що може бути випадково встановлений з оновленням системи або скомпрометування пам’яті пристроїв.

Службовий рівень. Перед введенням в дію та під час розробки КФС потрібно забезпечення безпечного програмування, вчасного та розумного тестування програмного коду, безпеки і правильності роботи. При розробці потрібно враховувати всі принципи безпечного проектування та можливість динамічного оновлення системи. Також необхідно забезпечити управління ідентифікацією та аутентифікацією користувача.

2.2 Аналіз можливих загроз для КФС

Для проведення детальної оцінки ризиків та змістовного врахування всіх можливих загроз для КФС, необхідно провести загальний аналіз КФС та визначити можливі вразливі місця системи, сприятливі для проведення атаки. Такі місця в системі називаються точками входу в систему. За допомогою аналізу вразливостей можна визначити потенційні точки входу, та виявнити, як порушник може використати перевагами цих вразливостей для впровадження шкідливих атак в систему.

2.2.1 Ієрархічний аналіз вразливостей

На рисунку 2.1 концептуально проілюстровано архітектуру типової КФС. Така архітектура описується у вигляді три-шарової схеми, яка складається з таких шарів, як кібершар, кіберфізичний шар та фізичний шар [37].

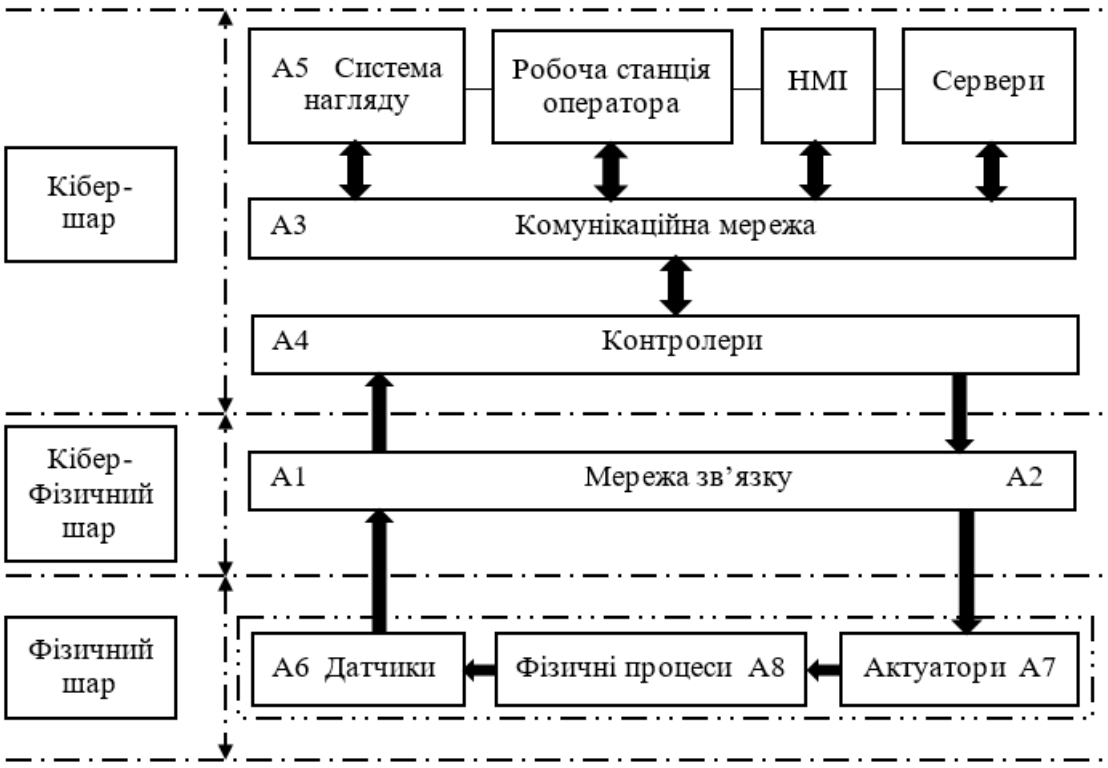


Рисунок 2.1 – Архітектура та склад КФС

У кібершарі знаходиться високорівневий інтерфейс типу людина-машина, алгоритми управління, пристрої які обробляють інформацію і дані. Його функції включають обробку даних, керування генерацією команд, а також управління та оптимізацію процесів на високому рівні [38].

Фізичний шар складається з датчиків, виконавчих механізмів та фізичних процесів, що зазвичай являють собою апаратну форму.

Кібрерфізичний шар – це вбудована мережева інфраструктура, яка полегшує обмін даними між фізичним та кібер рівнями. Для забезпечення безперебійної взаємодії використовуються протоколи зв'язку [39].

Для оцінки і аналізу безпеки КФС, а також безпеки потрібно, щоб було задіяно разом усі три шари, тобто має відбуватися обробка даних у кібершарі, передача даних у кіберфізичному шарі та команди управління і вимірювання датчиків на фізичному шарі.

2.2.2 Аналіз потоку даних

У КФС існує два типи потоку даних у циклі управління: потік даних датчика та потік даних виконавчого механізму. У КФС кіберсистема взаємодіє з фізичною системою, зчитуючи дані датчика та надсилаючи команди управління за допомогою кіберфізичної взаємодії. Потік даних датчика та потік даних виконавчого механізму взаємозалежні, зміна однієї сторони призведе до змін іншої сторони [40].

Потік даних датчиків – це значення вимірювання датчиків з фізичної системи. Скомпрометувавши дані датчиків, можна легко обманути контролери, що в свою дію призведе до виконання ними помилкових команд управління, що призведе до порушень безпеки фізичної системи.

Потік даних виконавчого механізму передається від кібершару до фізичної системи. В кіберпросторі у руках порушника така інформація може викликати небажані відхилення в роботі КФС.

2.2.3 Потенційні точки входу для атак в КФС

На рисунку 2.1 точки входу кібератак позначено як A1-A8. У таблиці 2.1 описана коротка характеристика даних атак. Атаки, які можуть бути ініційовані з точок A1, A2 і A3 націлюють на дані вимірювань датчиків і команди управління. За допомогою таких атак противники можуть перервати зв’язок, підслуховувати для отримання інформації про стан системи або підробляти передані пакети даних [20]. Такі дії можуть призвести до: атак відмови в обслуговуванні (DoS), обманних атак, атак введення помилкових даних та відтворювальних атак. Порушники можуть приховати інші нелегітимні дії від операторів, людини або алгоритмів виявлення подій, реалізованих в системі нагляду. Такі атаки називаються прихованими. Атаки, ініціалізовані на точках A4 та A5, можуть поставити під загрозу роботу контролера, всієї системи контролю або ж можуть змінити деякі конфігурації системи [20]. Атаки, розпочаті з точок A6, A7 і A8, можна розглядати як фізичні атаки.

Таблиця 2.1 – Потенційні точки входу атак в КФС

КФС шари	Позначення	Точки входу	Завдання	Сценарії атаки
Кібер-фізичний шар	A1	Мережа зв'язку між датчиками та контролерами	Перервати зв’язок між датчиками та контролерами; Маніпулювати або / та підслуховувати дані вимірювань, що надсилаються контролерам	Denial of Service (DoS) attack Deception attack False-data injection attack Replay attack
	A2	Мережа зв'язку між контролерами та приводами	Перервати зв’язок між контролерами та приводами; Маніпулювати або / і підслуховувати пакет даних, що надсилається виконавчим механізмам	Denial of Service (DoS) attack Deception attack False-data injection attack Replay attack
Кібер-шар	A3	Мережа зв'язку між контролерами та системами нагляду	Перервати зв’язок між контролерами та системами нагляду; Маніпулювати або / та підслуховувати пакет даних між контролерами та системами нагляду	Denial of Service (DoS) attack Deception attack False-data injection attack Replay attack
	A4	Контролери	Перервати нормальні операції керованого процесу, керувати логікою управління в контролерах або надсилати підроблені дані в систему нагляду / систему виявлення	Denial of Service (DoS) attack Deception attack False-data injection attack Ladder logic bombs
	A5	Система нагляду	Зміна конфігурації системи нагляду/виявлення порушень	Malware, code or program injection
Фізичний шар	A6	Фізичні процеси	Фізична атака на фізичні процеси	Direct physical attacks
	A7	Сенсори	Фізична атака на сенсори	Direct physical attacks
	A8	Актуатори	Фізична атака на актуатори	Direct physical attacks

Атаку можна назвати успішною, якщо вона задовольняє дві умови:

- Мета атаки досягнута;
- Атака залишається замаскованою до моменту досягнення мети.

Враховуючи мету атаки, порушник використовує наявні ресурси для здійснення послідовності шкідливих дій. Успішна атака може бути позначена дією або діями, які виводять фізичний процес за межі його безпеки, залишаючись не виявленим. Це визначення можна використовувати для оцінки успішності атаки чи ні [19]. Невдала атака – це атака яка була виявлена до моменту спричинення проблем безпеки в змінних процесу.

2.3 Опис теоретичного дослідження. Вибір методів

Перед проведенням оцінки ризиків КФС рекомендується провести тестування роботи датчиків, для уникнення технічних неполадок, вразливостей. Результати тестування будуть наведені в розділі 3.

Для оцінки ризиків досліджуваної КФС було запропоновано використання комбінації методів FoMRA та CRAMM. Для отримання кількісного результату оцінки ризиків використаємо спрощену формальну модель оцінки ризиків.

2.3.1 Спрощена формальна модель аналізу ризиків (FoMRA)

Ель Фрей та ін. у своїй роботі [36] розробили новий метод оцінки та управління ризиками FoMRA. FoMRA – це формальна модель оцінки ризиків, що базується на досвіді експертів, що створили метод MEHARI, і подібна за деякими параметрами до методу CRAMM. Метод формальної моделі оцінки ризиків відповідає рекомендаціям OECP та вимогам стандартів безпеки інформаційних систем ISO / IEC.

Даний метод можна представити у вигляді математичної моделі для отримання кількісної оцінки ризиків, чим може похвастатись не кожен метод.

Нехай A – Набір деяких активів:

$$A = \{ a_i : i = 1, \ldots, n_A \} \quad (1)$$

Додатково потрібно розглянути такі кінцеві множини:

$V = \{ v_j : j = 1, \ldots n_V \}$ - набір класів вразливості, (2)

$T = \{ t_k : k = 1, \ldots n_T \}$ - набір класів загроз, (3)

$S = \{ s_l : l = 1, \ldots n_S \}$ - набір сценаріїв ризиків, (4)

$DP = \{ dp_s : s = 1, \ldots n_{DP} \}$ - набір заходів, що зменшують потенціал, (5)

$DI = \{ di_t : t = 1, \ldots n_{DI} \}$ - набір заходів, що зменшують вплив. (6)

Множини, що були наведені вище показують класи системних активів, вразливості, класи загроз для активів, сценарії ризиків та загроз, і заходи, що дають змогу зменшити наслідки і потенційні можливості спричинені втратами активів загроз.

Нехай існує заданий і впорядкований набір $\Re$ *n*-значень, для аргументів в системі SI [36], який відповідає значенням множин *A, V, T, DP, DI* та *M, W*, де *M* і *W* – наступні значення та масиви, які дозволяють зменшити загрози, ризики і їх наслідки.

У множині $\Re = [r_{\min}, r_{\max}] \subset N$, де N набір натуральних чисел визначено додаткові допоміжні функції:

- Значення $A \to \Re^* \times \Re^* \times \Re^*$ - де $a \in A$ набуває значення: конфіденційність, цілісність і доступність.
- Значення $V \to \Re^* \times \Re^* \times \Re^*$ - отримує параметри: помилка, аварія, навмисно, значення яких залежить від природних вразливостей $v \in V$.
- Для значень A і V, множина значень $\Re^* = \Re + \{null\}$, де $null \in \Re^*$ є нейтральним значенням, це означає, що значення аргументу даної функції не має визначеної ознаки, отже йому не можна призначити якесь значення.
- Значення $T \to \Re$ - означає присвоєння заданої загрози $t \in T$ значенню t.
- Значення $DP \times S \times N \to \Re$ - це присвоєння заданих заходів, які зменшують потенціал загрози $d_p \in DP$ до значення d_p.

Крім того, для того, щоб отримати значення ризику $W^{s,a}$, для будь-якого сценарію ризику призначеного активу визначено такі масиви:

- Масив зменшення потенціалу $M^{s}_{pot\ n\times n\times n}$, який робить заявлене значення мурою зменшення потенціалів $W^{s}_{pot}[i,j,k]\in\Re$ залежно від значення $CM_{s,j}$, зокрема для $j=dp_1, \ldots, dp_{n_{DP}}$ і значення вразливості V(v).
- Масив впливу $M^{s}_{imp\ n\times n\times n}$, для $j=di_1, \ldots, di_{n_{DI}}$ робить заявлене значення вимірювального впливу $W^{s}_{imp}[i,j,k]\in\Re$, що залежить від $CM_{s,j}$.
- Масив зменшення впливу $M^{s,a}_{imp\ n\times n}$, який робить заявлене значення заходів, що зменшують вплив $W^{s,a}_{imp}[i,j]\in\Re$, залежить від значення W^{s}_{imp} , визначеного в масиві $M^{s}_{imp\ n\times n\times n}$, та значення вартості активу A(a).
- Масив ризиків $M^{s,a}{}_{n\times n}$, що встановлює заявлене значення ризиків $W^{s,a}[i,j]\in\Re$, залежно від значення W^{s}_{pot} ,що визначається з масиву $M^{s}_{pot\ n\times n\times n}$, та значення $W^{s,a}_{imp}$ яке можна отримати з масиву $M^{s,a}_{imp\ n\times n}$.

$CM_{s,j}\in\Re$ - це зважене значення міри, яке зменшує вплив і потенціал для певної загрози. Розрахунок значення потенціалу та дію впливу можна отримати з формули (7):

$$CM_{s,j}=\left\lfloor (r_{\max}-r_{\min})\cdot\frac{\sum R_i\times P_i}{\sum P_i}+r_{\min}+0.5\right\rfloor \qquad (7)$$

де:

- $j\in DP\cup DI$ - це реалізований захід чи контрзахід.
- $\lfloor x\rfloor$ - означає округлення результатів x до числа, яке належить множині $\Re$.
- R_i - це відповідь на запитання аудитора, може набувати значень 0 або 1.
- $P_i=value_X(j,s,no(R_i))$ - означає значення, що буде присвоєне для i-того питання, де X = DP або DI, що залежить від визначеного типу міри j у сценаріях s та кількості питань $no(R_i)$ пов'язаних з відповіддю R_i .

Значення визначених масивів M^{s}_{pot}, M^{s}_{imp} $M^{s,a}_{imp}$ та $M^{s,a}$, мають залежати від критичності процесів даної системи, і не повинні сприйматись жорстко. Критичність залежить від значення вразливості V(v), вартості активів A(a), та

ефективності впроваджених заходів, які зменшують потенційні можливості $DP(dp_s : s = 1,\ldots n_{DP})$, та удари $DI(dp_t : t = 1,\ldots n_{DI})$.

Також визначено для конкретних масивів такі набори масивів:

$$M_{pot} = \bigcup_{s\ :\ \exists_{dp \in DP}(s,dp) \in \overline{DP}} \{M^{s}_{pot}\} \quad (8)$$

$$M_{imp} = \bigcup_{s\ :\ \exists_{di \in DI}(s,di) \in \overline{DI}} \{M^{s}_{imp}\} \quad (9)$$

$$M^{a}_{imp} = \bigcup_{s\ :\ \exists_{di \in DI}(s,di) \in \overline{DI}} \{M^{s,a}_{imp}\} \quad (10)$$

$$M = \bigcup_{(a,s) \in \overline{A \times S}} \{M^{s,a}\} \quad (11)$$

2.3.2 Спрощена модель оцінки ризику CRAMM

Для якісної оцінки ризиків КФС буде доцільно використати один з етапів методу оцінки ризиків CRAMM. Даний метод продається у вигляді програмного забезпечення, і є доволі часозатратним, повна оцінка ризиків може зайняти навіть кілька місяців. Але для оцінки ризику нашої системи достатньо буде дотримуватись основних етапів та їх критеріїв, щоб отримати результати за відносно короткий термін.

Використовуючи метод CRAMM, для отримання значення ризику, потрібно враховувати, що це значення буде залежати від вартості активів, загроз та вразливостей КФС. На рисунку 2.2 зображено схему роботи методу CRAMM. Після проведення аналізу ризиків КФС буде розроблено перелік контрзаходів, які спрямовані на зниження, або усунення ризиків інформаційної безпеки системи.

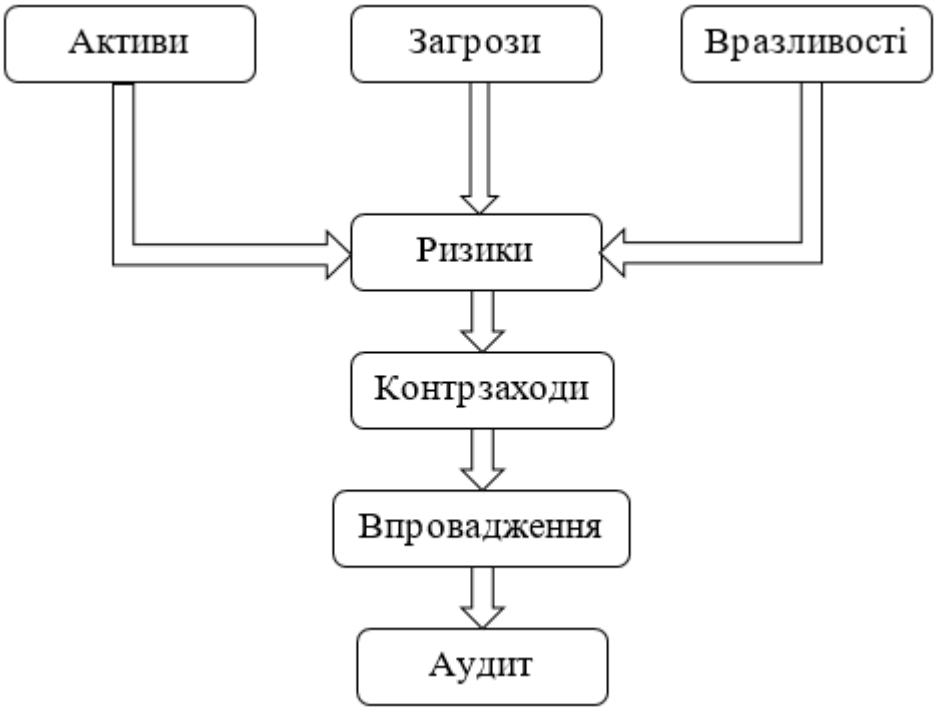


Рисунок 2.2 – Графічне представлення принципу роботи методу CRAMM

Для оцінки ризику нашої КФС потрібно обрати межі в яких ми будемо проводити аналіз. Для початку потрібно виділити основні складові безпеки за якими буде визначатися конкретна шкала абсолютного рівня загроз. У таблиці 2.2 наведено шкалу абсолютного рівня для кожної складової безпеки.

Таблиця 2.2 – Шкала абсолютного рівня

Складові безпеки					
Конфіденційність		Цілісність		Доступність	
Рівень	бали	Рівень	бали	Рівень	бали
Публічна	0	Низький	1-3	Низький	1-3
Обмежена	1-5	Помірний	4-7	Помірний	4-7
Конфіденційна	6-9	Високий	8-9	Високий	7-8
Захищена	10	Дуже високий	10	Дуже високий	9
-	-	-	-	Обов’язковий	10

Після призначення шкали для кожної складової безпеки, важливо визначити основні загрози для безпеки КФС для кожної складової безпеки. Для зручності представлення, в таблиці 2.3 буде наведено перелік загроз для кожної складової.

Таблиця 2.3 – Перелік загроз для основних складових безпеки

Складові безпеки	Конфіденційність	Цілісність	Доступність
Основні загрози	Витік інформації	Помилки введення	Збій у роботі КФС
		Спотворення інформації	Збої в роботі мережі енергопостачання
			Видалення інформації
			Збій мережі GSM зв'язку
	Отримання контролю над системою	Зміна показів датчиків	Фізичний, руйнівний вплив на апаратне забезпечення КФС
			Технічні порушення роботи датчиків.

Для кожної складової безпеки потрібно обов'язково вказати вимоги впливу і пріоритетність.

Оскільки в даній КФС інформація з обмеженим доступом не циркулює, то для «конфіденційності» можна призначити значення вимоги «5», що означає «обмежена» згідно вище заданої шкали.

Цілісність інформації, що циркулює в даній КФС є важливим критерієм, що видно з таблиці 2.3, тому для «цілісності» варто призначити значення «9», що означає «високий».

Проаналізувавши кількість основних загроз вказаних в таблиці 2.3 для «доступності» , можна впевнено призначити максимальне значення вимог до цієї складової, тобто значення «10» - «обов'язковий».

Для оцінки імовірності впровадження для кожної з загроз що були наведені в таблиці 2.3, буде доцільно створити шкалу вразливості, і для наглядності внести в таблицю 2.4.

Таблиця 2.4 – Шкала вразливості

№ з/п	Рівень вразливості	Значення (0-10)
1	Відсутній	0
2	Низький	1-4
3	Помірний	5-7
4	Високий	8-9
5	Дуже високий	10

Для оцінки рівня загрози потрібно провести просте обчислення. Значення рівня загрози *Рз* (форм.12) буде дорівнювати добутку вимог впливу *Вв* та рівня вразливості *Рв*.

$$Рз = Вв \cdot Рв \quad (12)$$

Отримане значення буде коливатися в межах від 0 до 100, тому доцільно скласти шкалу, за якою буде зручно визначати рівень ризику за допомогою оцінки загроз. Для зручності опишемо шкалу у вигляді таблиці 2.5.

Таблиця 2.5 – Шкала рівнів загрози

№ з/п	Рівень загрози	Значення (0-100)
1	Низький	0-33
2	Середній	34-67
3	Високий	68-100

Дана методика дозволяє визначити для кожної загрози відповідний рівень ризику. Після проведення всіх цих етапів отримані результати необхідно внести до таблиці і проаналізувати. Після чого для отриманих результатів оцінки ризиків обираємо контрзаходи для кожного ризику. Наступним етапом, як видно на рисунку 2.2 буде впровадження контрзаходів до системи, і після того проведення аудиту.

2.4 Опис спроєктованої КФС для оцінки ризиків

Для проведення теоретичних і експериментальних досліджень було спроєктовано КФС «Система контролю периметру» [22].

КФС призначена для забезпечення контролю безпеки периметру [41] від проникнення сторонніх осіб, порушників, злочинців.

В основі роботи системи лежить плата Ардуіно запрограмована для роботи з датчиками руху, та GSM-GPRS модулем для сповіщення користувача про вторгнення на контрольовану зону. У системі передбачено доступ до приміщення і до включення системи за допомогою RFID [42, 43]. Запрограмована смарт карта дозволяє вмикати сигналізацію при виході з приміщення, тобто з такою послідовністю :

1. Відкрив двері;
2. Приклав карту до зчитувача;
3. Закрив двері;
4. Сигналізація увімкнена.

Для вимкнення сигналізації потрібно прикласти карту до зчитувача перед відкриттям дверей. Якщо не скористатися картою при відкритті дверей, то спрацює сирена, а також для власника системи прийде смс-сповіщення про вторгнення. Таке ж повідомлення прийде власнику, якщо у приміщенні датчики руху помітять будь-який рух.

Для того, щоб увімкнути чи вимкнути сигналізацію віддалено, потрібно мати телефон, номер якого прописаний в статусі адміністратора в прошивці системи, та зателефонувати на номер, який встановлено в модуль системи. Для

увімкнення сигналізації потрібно подзвонити на номер системи і ввести значення «1». Для вимкнення системи потрібно ввести значення «2». Для відкриття електро-замка потрібно ввести значення «4». Для отримання інформації про значення поточної напруги, що живить систему потрібно ввести значення «9». При порушенні або втраті живлення – система відправляє повідомлення про відсутність живлення, або низький заряд. На рисунках 2.3 та 2.4 зображено функціональну схему роботи системи контролю периметру, як зі сторони користувача, такі і зі сторони самої системи.

Рисунок 2.3 – Функціональна схема роботи системи при віддаленому керуванні

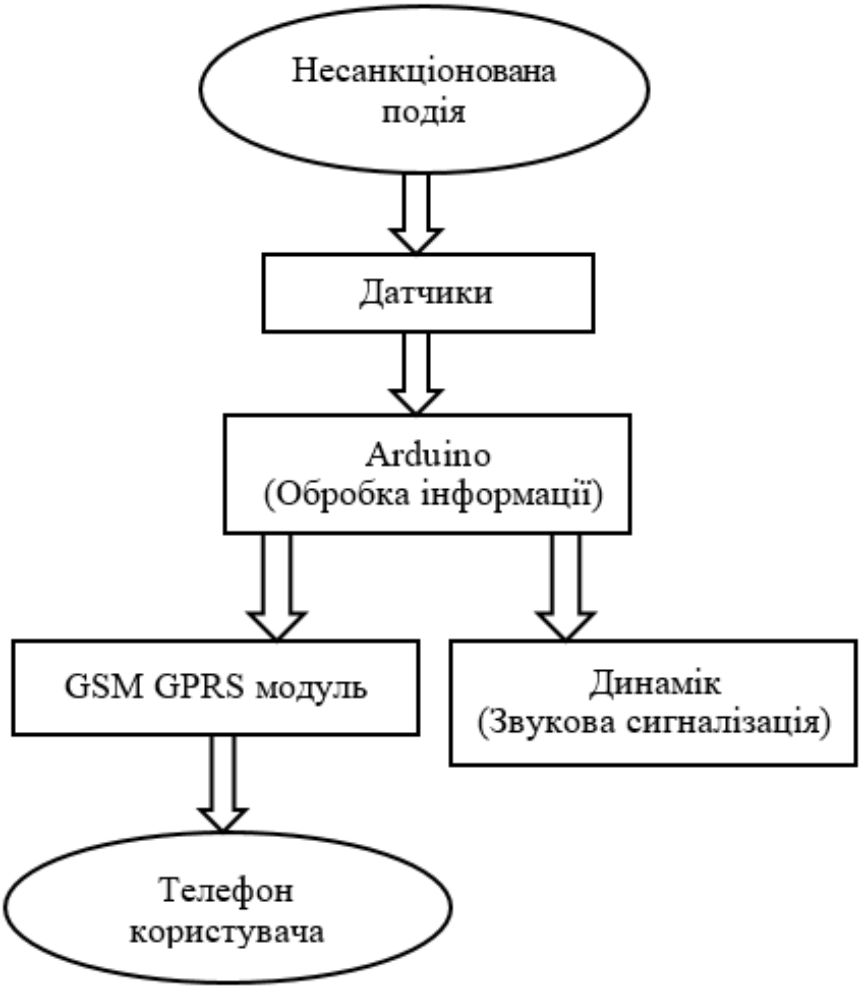


Рисунок 2.4 – Функціональна схема роботи системи під час виявлення датчиками несанкціонованих дій

Досліджувана у цій кваліфікаційній роботі реалізація даної системи є доволі бюджетною і зручною для користування як у будь-якій сфері, так і в своєму власному домі.

Для розробки КФС було використано середовище візуального програмування для Arduino FLProg-6.3.1 та середовище програмування Arduino.ide 1.8.12. Програмний код (скетч) прошивки системи збережено в додатку А.

Для проєктування і подальшої роботи КФС було використано наступне апаратне забезпечення:

- Плата Arduino Uno R3;
- Модуль зарядки на TP4056 з захистом від розряду;
- DC-DC підвищуючий MT3608 модуль живлення;

- Батарея Li-Ion 18650 3,7V 4200mah;
- Датчик руху/присутності HC-RS501;
- GSM GPRS модуль SIM800L;
- RFID MFRC – 522.

Схема підключень даних компонентів системи до плати Arduino Uno знаходиться у додатках Б і В.

На основі наведеного опису КФС, їх характеристик та програмного забезпечення було проведено дослідження, результати якого викладені в 3 розділі.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Тестування точності показів датчиків

Для безпечної роботи системи необхідне проведення тестування якості як самої КФС, так і її окремих складових, отримані дані з яких займають ключову роль у роботі системи.

В основі роботи КФС лежить пірометричний датчик присутності HC-RS501. На основі показів датчика система виконує своє головне завдання – сповіщає власника про несанкціоновану присутність або рух на контрольованій території. Перед введенням системи в дію потрібно провести тестування роботи датчиків. Тестування проводиться в кілька етапів. Першим етапом буде проведення перевірки роботи датчика присутності незалежно від всієї КФС, що далі і буде описано. Другий етап буде включати в себе тестування всієї системи, разом з всіма компонентами.

Для проведення досліду нам необхідні такі елементи:

- Плата Arduino UNO;
- Датчик руху HC-RS501;
- 1 світлодіод;
- USB кабель;
- 3 з’єднуючих проводи для датчика і плати;
- Комп’ютер для налаштування;
- Середовище програмування Arduino.ide;
- Скетч з налаштуваннями.

Спочатку необхідно ознайомитися з технічними характеристиками датчика, для детальнішого розуміння. У таблиці 3.1 наведено технічні характеристика датчика.

Таблиця 3.1 – Технічні характеристики HC-RS501

Пареметри	Знчення
Напруга живлення	4.5-20 В
Струм споживання	50 мА
Напруга на виході OUT	HIGH - 3,3 В,
	LOW - 0 В
Інтервал виявлення	3-7 м
Тривалість затримки після спрацьовування	5 - 300 с
Кут спостереження	до 120°
Час блокування до наступного вимірювання	2.5 с
Режими роботи	L - одиночне спрацьовування,
	H - спрацьовування при кожній події
Робоча температура	від -20° до + 80° С
Габарити	32x24x18 мм

Ознайомившись з характеристиками потрібно відкалібрувати параметри спрацювання. Налаштувати дистанцію виявлення та затримку можна за допомогою двох потенціометрів, які встановлені на датчику. Також необхідно вибрати режим роботи. Режим роботи налаштовується за допомогою перемички на платі. У режимі одиночного спрацювання, при виявленні руху, параметр OUT налаштовується на високий рівень сигналу на період затримки, що встановлений за допомогою потенціометра. Під час цього датчик не реагує на рухомі об’єкти. Такий режим зручний для використання в охоронних системах та подачі сигналу тривоги. При налаштуванні роботи датчика на другий режим, датчик буде реагувати на кожне виявлення руху. Такий режим краще використовувати для побутових цілей, наприклад автоматичного включення освітлення.

Необхідно зауважити, що після ввімкнення модуля буде проводитись його автоматичне калібрування, що займе приблизно одну хвилину. Після цього

датчик буде готовий до роботи. Також задля правильності роботи датчика, необхідно його встановлювати подалі від відкритих джерел світла.

Після ознайомлення з характеристиками та налаштуванням параметрів, можна підключати датчик до плати. Для цього як показано на рисунку 3.1, потрібно підключити контакт GND (земля) датчика до відповідного контакту GND плати, живлення датчика підключаємо до контакту 5В, а контакт VCC датчика до цифрового порта 2. Після цього можна підключати світлодіод, як показано на рисунку 3.1.

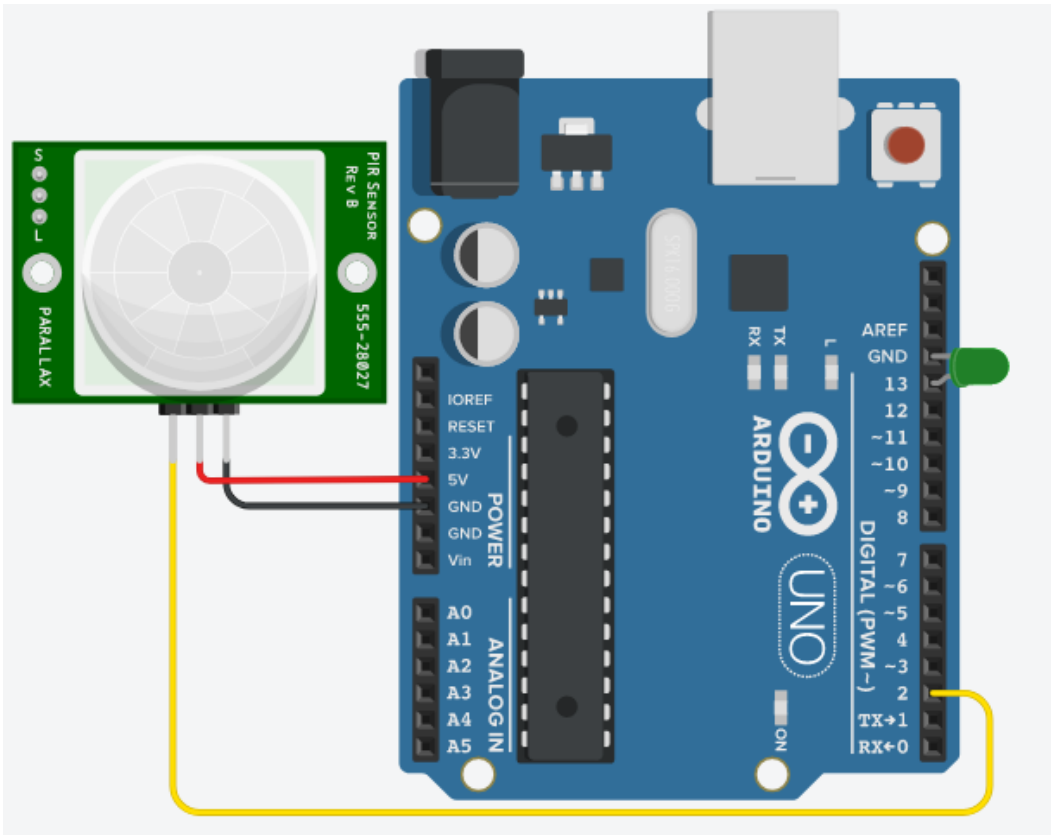


Рисунок 3.1 – Схема тестового підключення датчика

Коли все підключено правильно, можна підключати плату до комп’ютера за допомогою USB-кабеля. За допомогою середовища розробки Arduino.ide завантажуємо скетч на плату. Лістинг тестового скетчу знаходиться в додатку Г.

Скетч використовує вбудований діод плати, тому після підключення світлодіода в 13 цифровий порт, він буде засвічуватись при кожному виявленні руху датчиком.

Відкривши монітор послідовного порту можна спостерігати чотири значення, які зчитуються з цифрового виводу (помножені на 5), а також верхні і нижні пороги напруги компаратора (рис.3.2).

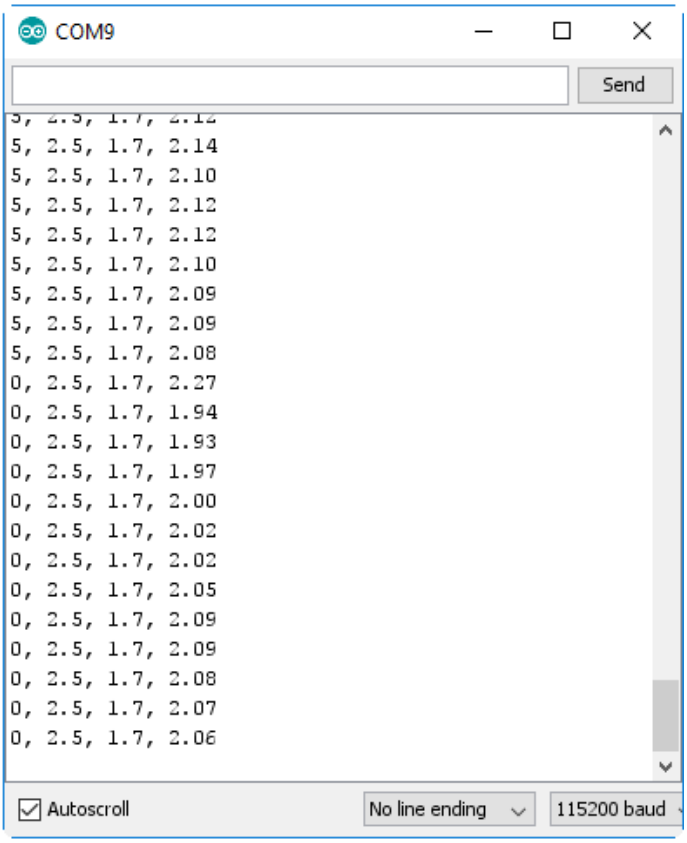


Рисунок 3.2– Значення з монітору послідовного порту

Оскільки візуалізація цифрового виходу та (при можливому підключені до) аналогового виходу, датчика за допомогою послідовного терміналу може бути важкою, за допомогою послідовного плотера можна отримати певну залежність у вигляді графіка (рис.3.3).

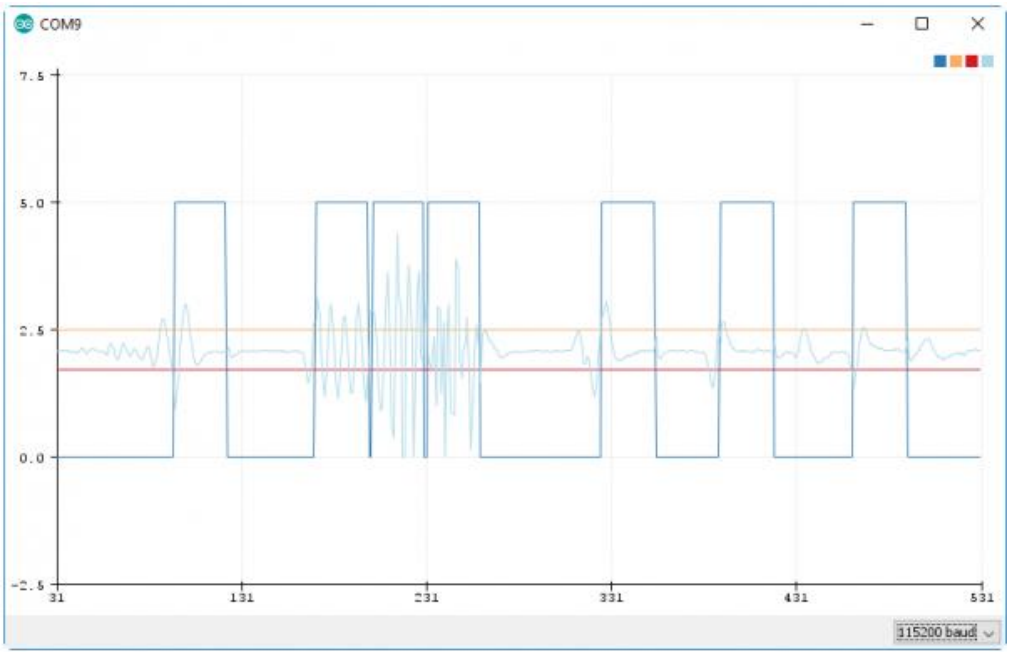


Рисунок 3.3 – Монітор послідовного плотера

Темно синій графік показує значення цифрового виводу, світло сині значення – аналогові. Помаранчева та червона прямі лінії представляють верхній та нижній пороги, які аналогове значення має перевищувати, щоб викликати рух.

Для проведення експерименту необхідно помахати рукою біля датчика, щоб краще відчути поведінку аналогового виходу. Як видно з рисунку 3.4, енергійний рух перед датчиком призводить до великої стрибкоподібної хвилі.

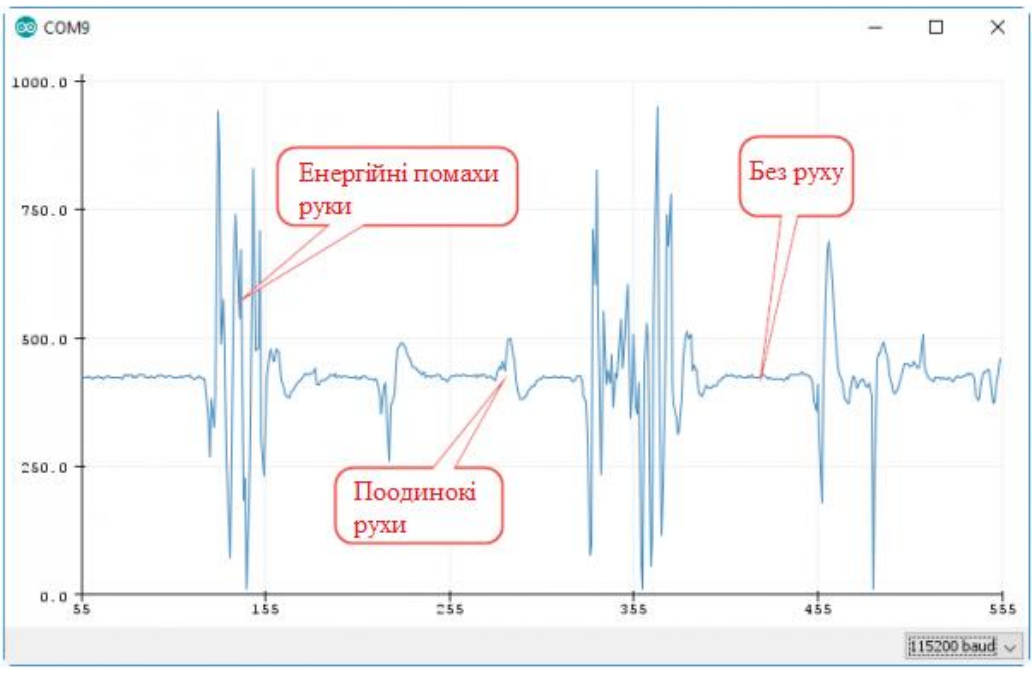


Рисунок 3.4 – Графік значень руху отриманих датчиком

Як видно з результатів проведеного експерименту датчик успішно пройшов тестування і готовий до експлуатації в КФС.

3.2 Оцінка ризиків методом FoMRA

У другому розділі нами було розглянуто принцип роботи та математичну модель формалізованого методу оцінки та управління ризиками FoMRA. Третій розділ оперує даними, які є результатами відповідних дослідів, у цьому підрозділі буде проведено оцінку ризиків КФС методом FoMRA.

Першим етапом для отримання кількісної оцінки за допомогою даного методу необхідно визначити шкалу значень ризику. У таблиці 3.2 наведено шкалу значень ризику інтерпритовану відносно шкали методу оцінки ризиків CRAMM.

Таблиця 3.2 – Шкала значень ризику FoMRA відповідно до CRAMM

Позначення ризику	FoMRA	CRAMM
Незначний ризик	1	1, 2
Допустимий ризик	2	3, 4
Неприпустимий ризик	3	4,6
Нестерпний ризик	4	7

Крім того необхідно визначити деяку шкалу класифікацій для елементів системи. У таблиці 3.3 і таблиці 3.4 представлено систему класифікації для активів, та систему класифікації для вразливостей або загроз відповідно.

Таблиця 3.3 – Система класифікації активів

Рівень активу	Значення рівня активу
1	Менш важливий
2	Важливий
3	Дуже важливий
4	Критичний

Таблиця 3.4 – Система класифікації вразливостей

Рівень вразливості	Значення рівня вразливості
1	Дуже низький / низький
2	Середній
3	Високий
4	Дуже високий

Наступним етапом буде проведення попереднього аудиту системи. Дані аудиту будуть зазначатись відносно визначених значень та записуватись в таблицю 3.5 для ресурсів визначено параметри конфіденційність, цілісність, доступність, та таблицю 3.6 для вразливостей відповідно аварія, помилка, навмисно (Accident, Error, Voluntary).

Таблиця 3.5 – Результати попереднього аудиту ресурсів

A	Активи	Значення A(a)		
		Конфіденційність	Цілісність	Доступність
a_1	Прикладне програмне забезпечення, пакети або проміжне програмне забезпечення (виконуваний код)	1	2	3
a_2	Дані про стан фізичного середовища отримані датчиками	null	4	2
a_3	Мережа зв'язку	3	null	4
a_4	Ключі доступу, смарт-карти	2	null	4
a_5	Апаратне забезпечення системи	null	2	4

Таблиця 3.6 – Результати попереднього аудиту вразливостей

V	Вразливості	Значення V(v)		
		Аварія	Помилка	Навмисно
v_1	Помилки програмного забезпечення	null	3	null
v_2	Помилки введення та спотворення інформації	null	null	2
v_3	Зміна показів датчиків	null	null	3
v_4	Витік ключової інформації	null	null	2
v_5	Отримання НСД до системи	null	null	3
v_6	Збої роботи КФС	4	null	null
v_7	Збої в роботі мережі живлення	4	null	null
v_8	Збої в мережі зв'язку	4	null	null
v_9	Видалення інформації	null	null	4
v_{10}	Порушення та збої в роботі давачів	null	4	null
v_{11}	Фізичний вплив на апаратне забезпечення КФС	null	null	4
v_{12}	Втрата або підробка смарт-карти	null	3	null

Для кожного зі сценаріїв загрози (s_1, s_2, ..., s_n) було розраховано значення ризику $W^{s,a}$. Для обчислення значення $W^{s,a}$ використовувались результати аудиторської анкети.

Аудит стосувався впроваджених стримуючих заходів, запобігання потенційним загрозам (заходи, визначені у формулі 5) та захисних, профілактичних, а також паліативних заходів, залежно від виду загрози (заходи, що зменшують загрозу, формула 6).

У додатку Д наведено блок запитань для анкетування і проведення аудиту стримуючих заходів для запобігання та попередження можливих загроз.

Значення відповіді у таблиці варіює між 0 та 1, що означає «ні», «так» відповідно.

Відповідно до формули 7, ми розраховуємо зважене значення мір $CM_{s,j}$. Зі значень вказаних у формулі $CM_{s,j} = 1$ означає, що міра неефективна, а $CM_{s,j} = 4$ означає, що вона дуже ефективна для кожного виявленого сценарію.

Далі буде проведено обчислення $CM_{s,j} = dp_1$ для заходів, взятих з таблиці додатку Д для обраного сценарію s_i:

$$CM_{s,j=dp_1} = \left[(4-1) \times \frac{(1 \cdot 4 + 1 \cdot 3 + 1 \cdot 4 + 1 \cdot 4 + 1 \cdot 4 + 1 \cdot 2)}{35} + 1.5\right] = 3.3$$

Значення міри можна округлити до меншого, що буде рівне «3», і це означає що міра впливу є ефективною і навіть у деяких випадках дуже ефективною для кожного сценарію загрози.

Оскільки КФС, яку ми аналізуємо не містить конфіденційної інформації, або інших даних, викрадення яких може завдати матеріальних збитків, то в пріоритеті буде оцінка ризиків щодо доступності. Отже виходячи з вище описаного твердження, можна сказати, що сценарії ризиків орієнтовані на забезпечення доступності.

У додатку Е наведено результати розрахунку зважених значень $CM_{s,j}$ для можливих загроз. Як видно з таблиці, деякі показники $CM_{s,j}$ мають значення, рівне 1. Це значення може бути результатом обчислення, як зазначене вище, або може бути прийняте довільно за відсутності таких заходів.

Таблиці охоплюють усі параметри безпеки та типи активів з використанням вразливостей у межах FoMRA. Під час аналізу були використані анкети аудиту відповідно до вимог безпеки.

Подальшою процедурою буде $W_{pot}^{s}, W_{imp}^{s}, W_{imp}^{s,a}$ та обрахунок $W^{s,a}$, що згадуються вище. Розраховані значення ризиків за допомогою FoMRA для вибраних сценаріїв для зручності наведено в таблиці 3.7.

Таблиця 3.7 - Розраховані значення ризиків за допомогою FoMRA

S	$CM_{s,j=dp1}$	$CM_{s,j=dp2}$	W^{s}_{pot}	W^{s}_{imp}	$W^{s,a}_{imp}$	$W^{s,a}$
s_1	2	1	2	2	2	2
s_2	3	2	3	3	2	2
s_3	1	1	1	1	4	4
s_4	3	4	3	3	2	4
s_5	3	3	3	2	3	3
s_6	1	4	3	3	4	4
s_7	1	3	2	2	2	3
s_8	2	3	2	2	2	2
s_9	2	1	2	2	2	3

Враховуючи додаткові значення активів, які були внесені до сценаріїв, що призвело до дублювання деяких з них відносно зміни активу, можна помітити деяку зміну значення міри і відносно цих значень значення загрози різко змінюється. Таку зміну значень можна побачити в таблиці з результатами оцінки ризиків (табл.3.7). Значення ризиків для більшості сценаріїв є критичними, але якщо впровадити заходи щодо їх послаблення, можна добитися помірних результатів, і отримати незначні значення ризику приблизивши ці значення до одиниці або двійки.

3.3 Метод оцінки ризиків CRAMM

Для якісної оцінки ризиків було використано один з найпоширеніших методів – CRAMM.

У другому розділі було описано етапи, та принцип роботи даного методу, за якими буде проведено оцінку ризиків спроєктованої КФС. У даному розділі будуть описані результати оцінки ризиків на основі даного методу, та їх обґрунтування.

Для початку потрібно описати основні загрози для нашої КФС на рівні конфіденційності, цілісності та доступності і занести дані для зручності у таблицю. У таблиці 2.3 наведено перелік основних загроз для досліджуваної КФС.

На основі отриманого переліку загроз для кожної складової безпеки потрібно обов'язково вказати вимоги впливу і пріоритетність.

Оскільки в даній КФС інформація з обмеженим доступом не циркулює, то для «конфіденційності» можна призначити значення вимоги «5», що означає «обмежена» згідно вище заданої шкали.

Цілісність інформації, що циркулює в даній КФС є важливим критерієм, що можна побачити на основі загроз з таблиці 2.3, тому для «цілісності» варто призначити значення «9», що означає «високий».

Проаналізувавши кількість основних загроз вказаних в таблиці 3.10 для «доступності» , можна впевнено призначити максимальне значення вимог до цієї складової, тобто значення «10» - «обов'язковий».

Описані вище значення для основних складових безпеки було занесено в таблицю 3.8.

Таблиця 3.8 – Вимоги впливу для складових безпеки КФС

Складові безпеки	Значення вимоги впливу
Конфіденційність	5
Цілісність	9
Доступність	10

Для переходу до наступного етапу, необхідно призначити кожній загрозі значення рівня вразливості. У розділі 2 в таблиці 2.5 було наведено шкалу вразливостей, значення якої ми призначимо для наших загроз. Для зручності буде доцільно занести дані в таблицю 3.9.

Таблиця 3.9 – Рівні вразливості загроз

Складові безпеки	Загроза	Рівень вразливості
Конфіденційність	Витік інформації	5
	Отримання контролю над системою	10
Цілісність	Помилки введення	7
	Спотворення інформації	8
	Зміна, спотворення показів датчиків	10
Доступність	Збій у роботі КФС	9
	Збої в роботі мережі енергопостачання	5
	Збій мережі GSM зв'язку	10
	Видалення інформації	8
	Фізичний, руйнівний вплив на апаратне забезпечення КФС	10
	Технічні порушення роботи датчиків та іншого апаратного забезпечення	9

Наступним етапом оцінки ризиків КФС буде оцінка рівня загрози. Щоб обрахувати значення рівня загрози треба знайти добуток вимог впливу на рівень вразливості. У додатку Є наведено отримані результати оцінки рівня загрози та рівня ризику для кожної з загроз.

Проаналізувавши отримані результати з додатку Є можна зробити цікавий висновок. Майже всі наведені загрози мають критично високий рівень ризику для проєктовуваної КФС. Також можна виділити, що у порівнянні з IT-системами, - КФС мають дещо іншу пріоритетність відносно основних складових безпеки. Як видно з додатку Є конфіденційність інформації для КФС

не є головним пріоритетом, як наприклад у IT-системах, де забезпечення конфіденційності інформації є головним завданням. Таку аналогію можна показати за допомогою таблиці 3.10, де наведено пріоритетність даних властивостей.

Таблиця 3.10 - Пріоритет між основними цілями безпеки в КФС та IT

Пріоритет	КФС	IT
Високий	Доступність	Конфіденційність
Середній	Цілісність	Цілісність
Низький	Конфіденційність	Доступність

На основі проведеної оцінки рівня ризиків можна переходити до наступного етапу методу оцінки ризиків CRAMM. Який являє собою розробку рекомендацій та контрзаходів щодо мінімізації або усунення можливості застосування виявлених ризиків для проведення атаки.

3.4 Контрзаходи та рекомендації для усунення виявлених ризиків

На попередньому етапі було проведено оцінку основних ризиків та загроз для «Системи контролю периметру». Метою даного підрозділу є розробка рекомендацій та контрзаходів для запобігання виявлених загроз і попередження нових.

Найбільш небезпечними ризиками для нашої КФС є такі як:

- Збій в мережах електропостачання, порушення живлення системи. Така загроза може тимчасово вимкнути систему, або ж взагалі пошкодити апаратну складову системи, від чого коректність роботи, або ж і сама робота системи стає під знак питання.
- Збої в мережах зв’язку. У нашому випадку збої в мережах GSM зв’язку, які унеможливлюють можливість віддаленого керування системою, а також унеможливлюють нормальне функціонування системи сповіщення про вторгнення і т.д.

- Руйнівний фізичний вплив на компоненти і датчики системи. Такий вплив може сприяти НСД до системи керування КФС зловмисником, а також вихід з ладу даної КФС.
- НСД до системи керування КФС за допомогою атак на смарт карти, а також перехоплення GSM сигналів.

Беручи до уваги вище перелічені загрози можна сказати, що у даній КФС найважливішим завданням є забезпечення доступності, а вже потім цілісності та конфіденційності.

Крім вище перерахованих загроз для нашої КФС існує ще чимало різних загроз роботі системи. У даному підрозділі будуть описані способи протидії ризикам описаних у підрозділі 3.4, а також методи та заходи протидії ризиками які є їх похідними, а також індивідуальним ризикам системи.

Для захисту системи від збоїв у системах електропостачання рекомендується використовувати системи UPS, стабілізатори вхідної напруги, а також модулі зарядки з захистом від високої напруги. При порушенні живлення системи, вимкненні системи електропостачання у описуваній системі контролю периметру використовується додаткова батарея, яка подає живлення для системи, у разі аварійного вимкнення мережі електропостачання. Також при таких збоях у мережі електропостачання програмно описана процедура, яка відправляє сповіщення власнику системи про порушення живлення. Також рекомендується встановити систему розумної зарядки, задля продовження роботи акомуляторів, а також постійного резервного забезпечення електроживлення для системи.

Щоб уникнути збоїв у роботі GSM мережі важливо використовувати сім-карту надійного оператора, з хорошим покриттям, а також розмістити антену в місці з високим покриттям і подалі від ока зловмисника.

Щоб захистити КФС від пошкодження компонентів та датчиків порушником, потрібно розмістити датчики в недоступних для порушника місцях. Обов’язково використовувати металеві короба для проводки кабелів. Компоненти які знаходяться за межами контрольованої зони, такі як модуль зчитування смарт карт або ж антену для модуля зв’язку потрібно захистити

спеціальними сейфами/кейсами/шафами, що надійно закриваються, і не допускають потрапляння вологи, та інших шкідливих чинників, а також обмежують можливості зловмисника.

Датчики руху/присутності потрібно розміщувати на стелі біля потенційно можливих ділянок потрапляння у приміщення, а також біля компонентів які ми захищаємо. Якщо дозволяє кошторис, рекомендується використовувати датчики у кількості 1 на 2.5 $м^2$ – це забезпечить максимальний контроль приміщення.

Сам мікроконтролер/плату варто зберігати в недоступних для злочинців місцях, наприклад в тих же самих настінних шафах з замком, які не дозволять проникнути до плати побічним сигналам, а також вбережуть від руйнівної дії зловмисників, а також природних негативних чинників, наприклад води.

Для уникнення атак на смарт карти потрібно надійно берегти саму карту, а в разі її втрати видалити з системи її ідентифікатори. Це вбереже систему від підробки смарт карти, і отримання доступу до системи і приміщення завдяки підробці.

Для віддаленого керування системою, важливою умовою є перелік номерів телефону збережених в системі для вводу команд, та відправлення сповіщень. Такий перелік повинен зберігатись в повному секреті, для запобігання підробки номеру телефону, що в свою чергу може дати доступ до керування системою.

Номер телефону самої системи повинен бути прихований. Така умова задовільняє властивість конфіденційності, і унеможливлює можливість підглядання номеру телефону системи, що відправляє сповіщення, а також отримує дзвінки з командами.

Рекомендовано ведення журналу, який моніторитиме всі підключення, введені команди, включення і виключення системи, покази напруги, заряд, покази датчиків, відкривання дверей, всі дзвінки, повідомлення, стан мережі зв’язку, а також номери телефонів, які намагались, або ж керували системою, час з’єднання, а також тривалість з’єднання.

Рекомендовано мати резервну копію для прошивки всієї КФС та для смарт карт. Такий метод дозволить відновити дані системи, і відновити функціонал у разі втрати, чи видаленні прошивки системи.

Система повинна забезпечувати вчасне та безперебійне сповіщення власника про будь-які зміни в системі, чи виявленні будь-якого втручання.

Також для забезпечення нормального функціонування системи, варто проводити вчасно і регулярно перевірку працездатності системи та її окремих компонентів. У разі виявлення поломок – вчасно проводити ремонт або заміну відповідних елементів. Регулярне очищення пам’яті пристрою та оновлення прошивки дозволить системі безперебійно працювати та дає можливість доповнення системи новими компонентами.

3.5 Обґрунтування отриманих результатів

Підсумовуючи результати вище проведених дослідів, можна зробити певні висновки. Проведена оцінка ризиків за допомогою методів FoMRA та CRAMM дала однакові результати, що показують, на вразливості системи на рівні доступності.

Метод CRAMM передбачає розробку контрзаходів для усунення чи послаблення кожної вразливості. Результати цього етапу у вигляді контр заходів були наведені в попередньому підрозділі. Після введення в дію контрзаходів система буде вважатися безпечною.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Як для кожного приладу є інструкція з експлуатації, так і на підприємствах є певні протоколи для роботи з тими чи іншими засобами, при конкретних умовах. У нашому випадку, відповідно до теми кваліфікаційної роботи – це буде застосування правил та протоколів у сфері охорони праці відповідно до чинних нормативних документів з охорони праці для безпечної роботи осіб, які працюють з комп'ютерними пристроями та працюють чи знаходяться поруч зі складовими апаратного забезпечення кіберфізичних систем. Після встановлення та введення в дію кіберфізичної системи, для безпеки працівників, які матимуть доступ до даної кіберфізичної системи, рекомендовано обов'язкове проведення організаційних заходів з охорони праці, таких як проходження інструктажу з техніки безпеки, обов'язкове навчання працівників протоколам роботи з КФС, та чітко прописані обов'язки відповідальних осіб.

4.1.1 Протипожежні заходи для кіберфізичних систем

Заходи направлення на попередження і усунення пожеж мають особливе місце серед технічних і комп'ютеризованих засобів і систем. Оскільки це в першу чергу має безпосередній вплив на людей що працюють даними засобами та системами.

Описана у попередніх розділах кіберфізична система являє собою систему контролю периметру, яка забезпечує безпечний доступ до контрольованої зони. Одною з основних властивостей кіберфізичних систем є доповнюваність, тобто дана система може бути доповнена додатковими елементами, датчиками, які дозволятимуть отримувати і аналізувати дані про фізичне середовище, і на основі цього аналізу виконувати певні дії, описані в протоколах обробки. Отже, можна без додаткових затрат укомплектувати систему пристроями та датчиками типу

протипожежних сповіщувачів та давачів, замість того, щоб додатково встановлювати цілу окрему систему протипожежної сигналізації.

Пожежну сигналізацію можна описати як сукупність пристроїв, які сповіщають про пожежну небезпеку, лінії до якої вони підключені, контроллер який обробляє отримані покази з сповіщувача та система зв'язку.

Принципом роботи пожежних сповіщувачів є перетворення даних про пожежу, тобто самих проявів пожежі в електричні сигнали, які передаються по каналах зв'язку до контроллера. Завданням контроллера є отримання інформації від пожежних давачів, та обробка отриманої інформації, залежно від значення якої буде задано завдання створення та передачі електричного сигналу про виникнення пожежу, або виду якоїсь несправності, до інших пристроїв, які призначені для вирішення цих питань.

Протипожежні датчики бувають таких видів [44]:

- Теплові. Вони отримують значення температури з фізичного середовища, та при перевищенні норми, або швидкості її підвищення передають ці дані на контроллер;
- Димові. Даний вид датчиків спрацьовує при виявленні в фізичному середовищі аерозольних продуктів горіння;
- Вогню. Датчики спрацьовують від дії електромагнітного випромінювання, джерелом якого є полум'я.

Залежно від характеристики контрольованої зони здійснюється вибір пожежних давачів і оповіщувачів. Також потрібно забезпечувати вчасну заміну протипожежних сповіщувачів у зв'язку з зношуванням діючих елементів сповіщувача. Сповіщувачі, давачі та інші апаратні засоби пожежної сигналізації і цілої КФС повинні розміщуватись відповідно до правил встановлених в [45]

Беручи до уваги те, що для гасіння пожежі найчастіше використовують воду, важливо забезпечити створення та працездатність протипожежних систем водооснащення. В основі протипожежних систем водопостачання лежить набір водонапірних споруд та пристроїв, що призначено для збору, зберігання та транспортування води до джерела займання. Метою даної системи є забезпечення необхідної кількості води, та необхідного напору, достатніх для

відповідного часу гасіння пожежі і при умові достатньої надійності всього комплексу системи водооснащення. Отже кожне підприємство мусить бути забезпеченим необхідними об’ємами води та установками, призначеними для пожежогасіння.

Також для забезпечення безпеки працівників від пожежі рекомендується обов’язкове використання первинних засобів пожежогасіння. До таких засобів відносять: вогнегасники, такий пожежний інвентар, як бочки з водою, ящики з піском, відра, лопати, сокири, ломи, гаки, покривала з негорючого полотна тощо. Первинні засоби доцільні при використанні для пожежогасіння на початкових стадіях розвитку.

Також слід враховувати важливість організаційних заходів пожежної безпеки. Відповідно до Наказу «Про затвердження Правил пожежної безпеки в Україні»[46] забезпечення пожежної безпеки організацій, установ та ін. – покладено на їх керівників і уповноважених осіб, якщо інше не передбачає відповідний договір.

Повноваження у галузі пожежної безпеки виробничих об’єднань у обов’язковому порядку мусять бути визначені договорами чи статутами даних об’єднань.

Особи, що порушують правила, норми, постанови посадових осіб органів пожежного нагляду, або перешкоджають їхній діяльності будуть притягнуті до відповідальності відповідно до чинного законодавства України.

Для роботи з кіберфізичною системою «Система контролю периметру» визначено основні правила та норми роботи, відповідно до «Примірної інструкції з охорони праці під час експлуатації електронно-обчислювальних машин»[47]. Система може використовуватись, як пожежна сигналізація, що врази додає зручності та безпеки використання.

Отже враховуючи виконання вище описаних заходів з охорони праці, та правил пожежної безпеки розробку можна вважати безпечною з точки зору охорони праці, що підтверджує [46]

4.2 Безпека в надзвичайних ситуаціях

Задля забезпечення правильної роботи будь-якої системи, не тільки кіберфізичної потрібно врахувати і оцінити всі можливі ризики, що можуть негативно вплинути на роботу такої системи. Оскільки для кіберфізичних систем найголовнішим завданням безпеки є забезпечення доступності інформації, потрібно забезпечувати правильну роботу апаратного забезпечення. Серед засобів технічного захисту інформації, та організаційних методів, також важливо забезпечити захист апаратури у екстремальних ситуаціях, тобто НС. У цьому підрозділі буде наведено та описано заходи захисту апаратного забезпечення кіберфізичної системи від впливу радіації.

4.2.1 Заходи та засоби направлені для зменшення деструктивних дій радіоактивного випромінювання на апаратне забезпечення кіберфізичних систем

Радіаційний вплив на складові частини апаратного забезпечення кіберфізичних систем залежить від різних чинників. Такими чинниками можна назвати: дозу радіації, вид випромінення, умови фізичного середовища та властивостей речовини, що була опромінена [48].

Для побудови радіо-електронної апаратури було використано такі елементи, які містять в своїй будові такі матеріали, як: органічні сполуки різного виду, наприклад смоли, діелектрики та ін., та напівпровідникові елементи, багато видів неорганічних матеріалів та металів. Найбільш чутливими до впливу радіоактивних випромінювань серед наведених вище матеріалів є метали. Така чутливість металів пояснюється наявною в них часто високою концентрацією вільних носіїв.

Від дії радіоактивного випромінювання на апаратне забезпечення кіберфізичних систем та РЕА, в апаратурі відбуваються певні процеси, які здатні негативно впливати на роботу компонентів схеми, що призведе до поломок апаратного забезпечення. Під час проходження потоку гамма-випромінення через елементи апаратного забезпечення КФС, відбувається виникнення вільних

носіїв електронів. Під час процесу переміщення вільних носіїв електричного заряду відбувається виникнення хибних імпульсів, що за певних умов можуть ввімкнути пристрої системи. У таблиці 4.1 наведено експозиційні дози $Д_I$ [48] випромінення, які можуть викликати зворотні зміни в елементній базі КФС.

Таблиця 4.1 – Максимальні значення експозиційної дози на КФС

Елементи апаратного забезпечення кіберфізичної системи	$Д_I$, Р
Діоди загального призначення	$10^4 \dots 10^6$
Транзистори	$10^4 \dots 10^6$
Мікросхеми	10^5
Інтегральні схеми	$5 \cdot 10^5$
Конденсатори	$10^7 \dots 10^9$
Резистори	$10^7 \dots 10^9$
Кварц	10^{10}

До впливу радіоактивного випромінювання найбільш чутливими можна вважати напівпровідникові елементи, фотоматеріали та оптичні засоби. Після наслідків деструктивних дій іонізуючого випромінювання у елементній базі кіберфізичних систем та радіо-електронної апаратури передбачено можливу заміну всіх експлуатаційних та електричних характеристик, що були пошкоджені, порушили свою структуру або зазнали змін після дії процесів іонізації. Після багаторічних практик та досліджень використання радіо-електричної апаратури в умовах дії деструктивного радіоактивного випромінення дозволяє зробити таку дефініцію:

1. При критичних рівнях радіоактивного випромінювання РЕА, а отже КФС може втратити свою працездатність.
2. Після проходження певного часу після отримання радіоактивного опромінення з рівнями нижчими ніж критичний рівень радіоактивного

випромінення, тобто $p_{гр}<p_{кр}$, можуть початись зворотні чи незворотні процеси в елементній базі РЕА, а отже і КФС.

Перший випадок для інженерної практики має найбільший інтерес. Під цим випадком розуміється аналіз та оцінення стійкості працездатності радіо-електонного обладнання, за умов її знаходження на території забрудненої радіоактивними елементами на період одної години після радіоактивного опромінення даної території.

Оцінювання стійкості роботи радіо-електронного обладнання відбувається в такій послідовності [49, 50]:

1. Провидиться аналіз РЕА, та визначення всіх елементів, що впливають на її роботу, це можуть бути наприклад: резистори, транзистори, мікросхеми тощо.
2. Для всіх елементів КФС та РЕА проводиться визначення значення максимально допустимої потужності які можуть мати дози гама-випромінення p_i або ж значення експозиційної дози. Отримані значення наведено в таблиці 4.2.

Таблиця 4.2 – Результати перевірки стійкості елементів РЕА та КФС

Елементи РЕА	$P_i, Р/с$	$Д_i, Р$	$p_{гр}, р/с\,; Д_{гр} Р$
Напівпровідники	p_1	$Д_1$	$p_{гр}\ (Д_{гр})$
Мікросхеми	p_2	$Д_2$	
Конденсатори	p_3	$Д_3$	

3. Проводиться аналіз даних наведених з таблиці 4.2, та обраховується значення межі стійкості $p_{гр}$ $(Д_{гр})$ роботи радіо-електронного обладнання відносно мінімального значення $p_1 Д_i$.
4. Відбувається порівняння граничного значення потужності $p_{гр}$ або ж граничного значення експозиційної дози $Д_{гр}$ з очікуваним p_{1max} $(Д_{max})$. Складаємо висновок про оцінку стійкості роботи.

$$\left.\begin{matrix} P_{гр} \geq P_{1max} \\ Д_{гр} \geq Д_{1max} \end{matrix}\right\} \text{– РЕА стійка до радіації;}$$

$$\left.\begin{matrix} P_{гр} < P_{1max} \\ Д_{гр} < Д_{1max} \end{matrix}\right\} \text{– РЕА нестійка до радіації;}$$

Гранично допустимий час роботи радіо-електронного обладнання за таких умов визначають на прикладі даних виразів:

$$t_{д} = \left(\frac{Д_{гр} \cdot К_{пос} + 1{,}33 \rho_{1max} \cdot \sqrt[4]{t_n^3}}{1{,}33 \rho_{1max}} \right)^{4/3}, \text{ год}$$

$$t_{д} = \left(\frac{Д_{гр} \cdot К_{пос} + 2 \rho_{1max} \cdot \sqrt[4]{t_n}}{2 \rho_{1max}} \right)^{2}, \text{ год}$$

5. На основі дефініції стійкості проводиться розроблення заходів щодо забезпечення стійкості радіо-електричного обладнання до радіації.

Заходи щодо забезпечення стійкості радіо-електричного обладнання до радіації.

Базуючись на дослідженнях українських та закордонних дослідників можна стверджувати, що при зміні параметрів радіообладнання можна підвищити працездатність в широкій вибірці доз радіоактивного випромінення. У 80% випадків є необхідність збільшення стійкості обладнання радіоактивного випромінювання. Для збільшення стійкості до радіо-випромінення рекомендується використовувати для розробки апаратури масивні апаратні екрани чи активний захист від дії радіоактивного випромінення. Також використовують схеми, що є малочутливими відносно зміни електро-параметрів, зниження напруги, зниження чутливості перемикальних схем при зміні вихідного сигналу та підвищення негативних заміщень на сітках газорозрядних приладів тощо.

Під час розробки даного розділу було проаналізовано завдання проведення заходів, які направлені на послаблення деструктивних дій радіоактивного випромінювання на обладнання КФС. Було проведено визначення елементів які можуть бути найбільш чутливими до радіації, а також розроблено заходи для забезпечення або підвищення рівня стійкості роботи КФС.

ВИСНОВКИ

У кваліфікаційній роботі було проведено аналіз теоретичної розробленості (1 розділ) у сфері кіберфізичних систем та методів оцінки ризиків. На основі результатів проведеного аналізу було обрано два найзручніших методів оцінки ризиків, які дозволять провести оцінку кіберфізичних систем. Цими методами стали CRAMM та FoMRA. Для проведення практичної оцінки ризиків було спроєктовано прототип кіберфізичної системи «Система контролю периметру».

У другому розділі було описано загальні підходи та принципи роботи обраних методів оцінки ризиків. На цій основі проаналізовано, виявлено та оцінено загрози інформаційним ресурсам, пов’язаним з КФС. Також у даному розділі було описано будову та принципи роботи «Системи контролю периметру» для попередження несанкціонованого доступу до інформаційних систем. Третій розділ дає відповіді на завдання, що були поставлені. Прототип пройшов тестування працездатності та якості апаратного та апаратного забезпечення, що дозволяє стверджувати, що він придатний для забезпечення захисту і функціонування інформаційно-телекомунікаційних систем з сучасними архітектурами та пов’язаними інформаційними потоками й процесами для внутрішніх і віддалених компонент. Також у цьому розділі було проведено оцінку ризиків системи обраними методами. На основі отриманих результатів оцінки ризиків, було зроблено такий висновок, що обидва методи виявили та оцінили рівні вразливостей однаково. В пріоритеті буде забезпечення доступності системи. Для отриманих ризиків було розроблено контрзаходи, при виконанні яких система може вважатися безпечною.

У четвертому розділі описано можливе використання прототипу у вигляді пожежної сигналізації. Для приведення системи до такого вигляду необхідно провести доповнення апаратного та програмного забезпечення спеціальними давачами та програмними командами. Також у четвертому розділі було описано заходи щодо зменшення впливу радіації на апаратне забезпечення кіберфізичних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ярощук І.В. Ризик-орієнтований підхід для розробки безпечних кіберфізичних систем на базі Arduino / І. В. Ярощук, Ю. Л. Скоренький // Збірник тез доповідей VIII Науково-технічної конференції „Інформаційні моделі, системи та технології“, 9-19 грудня 2020 року. — Т. : ТНТУ, 2020. — С. 73.

2. Xiaorong L. Safety and Security Risk Assessment in Cyber-Physical Systems / Xiaorong L, Yulong D , Shuang-Hua Y // Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, 2019, 14 p.

3. G. Wu “A survey on the security of cyber-physical systems,” Control Theory and Technology / G. Wu, J. Sun, J. Chen, vol. 14, 2016, 128 p.

4. NIST Special Publication 1500-201, Framework for Cyber-Physical Systems: vol. 1, Overview, 2017, 79 p.

5. Bush S.F. Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid, Wiley-IEEE Press, 2014, 570 p.

6. Characteristic, Architecture, Technology, and Design Methodology of Cyber-Physical Systems / C. Liu, F. Chen, J. Zhu, et al // Department of Computer Science & Technology, Anhui Normal University, Wuhu, Anhui, China, 2019, 46 p.

7. Tan Y., A concept lattice-based event model for cyber-physical systems. In: Proceedings of the International Conference on Cyber-Physical Systems / Tan Y., Vuran M., et al, 2010, 50-60 pp.

8. Krishna, V., Cyber physical internet. challenges, opportunities, and dimensions of cyber-physical systems / Krishna, V., Saritha, V., Sultana, P., 2015, pp. 76–97

9. Shmatko O. Development of methodological foundations for designing a classifier of threats to cyberphysical systems /O. Shmatko, S. Balakireva, A. Vlasov, N Zagorodna, O Korol, O Milov, O Petrov, S Pohasii, K Rzayev, V Khvostenko // Східно-Європейський журнал передових технологій, 3/9 (105), с. 6-19, 2020.

10. F. Zhang, Multi-layer datadriven cyber-attack detection system for industrial control systems based on network, System and Process Data / F. Zhang, H. A. D. E. Kodituwakku, W. Hines, J. B. Coble//IEEE Transactions on Industrial Informatics, 2019

11. Білостоцький Т. Математичне моделювання передачі даних в комп'ютерних мережах / Т. Білостоцький, Г. Осухівська // Матеріали II науково-технічної конференції „Інформаційні моделі, системи та технології", 25 квітня 2012 року — Т. : ТНТУ, 2012 — С. 36.

12. S. A. A. A. Cárdenas. Research challenges for the security of control systems / S. A. A. A. Cárdenas, S. S. Sastry // in Proceedings of the 3rd Conference on Hot Topics Security, Berkeley, pp. 1−6, 2008.

13. A. Singh. Study of Cyber Attacks on Cyber-Physical System / A. Singh, A. Jain, // 3rd International Conference on Advances in Internet of Things and Connected Technologies (ICIoTCT), 2018, p 686-690.

14. Lu, T. A New Multilevel Framework for Cyber-Physical System Security / Lu, T., Xu, B., Guo, X., Zhao, L., Xie, F. // 2013.

15. Ahmed, S. H. Cyber-Physical System: Architecture, applications, and research challenges / Ahmed, S. H., Kim, G., Kim, D. // In Wireless Days IEEE. IFIP, 2013, pp. 1-5.

16. Shi, J. A survey of cyberphysical systems. In Wireless Communications and Signal Processing (WCSP) / Shi, J., Wan, J., Yan, H., Suo, H. // International Conference on IEEE. 2011, pp. 1-6.

17. Zhang, L. Security threats and measures for the cyber-physical systems / Zhang, L., Qing, W. A. N. G., & Bin, T. I. A. N. // The Journal of China Universities of Posts and Telecommunications, 20, 2013, pp. 25-29.

18. Chris Northwood whilst, the University of York and University of Sheffield, Access mode: http://www.pling.org.uk/cs/cry.html,2010, Accessed Date:10 Nov 2020.

19. Ning X. Analysis, Design and Demonstration of Control Systems Against Insider Attacks in Cyber-Physical Systems / Ning X, The University of Western Ontario, 2019, 176 p.

20. Taormina R. Characterizing cyber-physical attacks on water distribution systems / R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld // Journal of Water Resources Planning and Management, 143(5): 04017009, 2017.

21. Гаваньо Б. І. Проблеми конфіденційності та безпекив кіберфізичних системах інтелектуальних будинків / Б. І. Гаваньо , Національний університет "Львівська політехніка" кафедра електронних обчислювальних машин, 2018, с 49-55.

22. Ярощук І.В. Можливості застосування пірометричних датчиків на платформі Arduino для контролю периметру / І. В. Ярощук, Ю. Л. Скоренький // Збірник тез доповідей III Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання“, 25 квітня 2020 року. — Т. : ТНТУ, 2020. — С. 16-17.

23. Evans M. Arduino in Action / M. Evans, J. Noble, J. Hochenbaum - Shelter Island, NY, 2013, 370 p.

24. IEC 62443: ‘Security for Industrial Automation and Control Systems’, 2013.

25. IEC 61511: ‘Functional safety - Safety instrumented systems for the process industry sector’, 2016

26. Обертинюк І. Л. Технології оцінки ризиків інформаційної безпеки відповідно до вітчизняних нормативних документів та міжнародних стандартів / І. Л. Обертинюк, О. В. Кареліна // Збірник тез доповідей VII Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 28-29 листопада 2018 року. — Т. : ТНТУ, 2018. — Том 2. — С. 132–133.

27. IEC 61508: ‘functional safety of electrical/electronic/programmable electronic safety-related systems’, 2010.

28. Modarres, M., and Cheon, S. W.: ‘Function-centered modeling of engineering systems using the goal tree–success tree technique and functional primitives’, Reliability Engineering & System Safety, 1999, 64, (2), pp.181-200.

29. Brissaud, F., Barros, A., Bérenguer, C., and Charpentier, D.: ‘Reliability study of an intelligent transmitter’, Proc. 15th ISSAT Int. Conf. Reliability and Quality in Design, San Francisco, United States, 2009, pp. 224-233.

30. Dunjó, J., Fthenakis, V., Vílchez, J. A., and Arnaldos, J., ‘Hazard and operability (HAZOP) analysis. A literature review’, Journal of Hazardous Materials, 2010, 173, (1-3), pp.19-32.

31. Kennedy, R., and Kirwan, B.: ‘Development of a hazard and operability-based method for identifying safety management vulnerabilities in high risk systems’, Safety Science, 1998, 30, (3), pp.249-274

32. Lee, D. A., Lee, J. S., Cheon, S. W., and Yoo, J.: ‘Application of system-theoretic process analysis to engineered safety features-component control system’, Proc. 37th Enlarged Halden Programme Group (EHPG) meeting, Storefjell, Norway, 2013.

33. Ebeling, C.E.: ‘An Introduction to Reliability and Maintainability Engineering’ (Waveland Press, Long Grove, Illinois, 1997, 2nd edn.2009).

34. Nourian, A., and Madnick, S.: ‘A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet’, IEEE Transactions on Dependable and Secure Computing, 2018, 15, (1), pp.2-13

35. Yazar Z. A Qualitative Risk Analysis and Management Tool – CRAMM / Z. Yazar - SANS Institute Information Security Reading Room, 2020, 14 p.

36. El Fray, I., Kurkowski, M., Pejas, J., Mackow, W.: A New Mathematical Model for Analytical Risk Assessment and Prediction in IT Systems. Control and Cybernetics 41(2012) 1-28

37. Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: Analysis, challenges and solutions," Computers & Security, vol. 68, pp. 81−97, 2017.

38. S. C. Genge B L, Fovino I N, et al., "A cyber-physical experimentation environment for the security analysis of networked industrial control systems," Computers & Electrical Engineering, vol. 38, pp. 1146−1161, 2012.

39. A. Ashok, "Attack-resilient state estimation and testbed-based evaluation of cyber security for wide-area protection and control," Iowa State University, 2017.

40. R. Akella, H. Tang, and B. M. McMillin, "Analysis of information flow security in cyber–physical systems," International Journal of Critical Infrastructure Protection, vol. 3, pp. 157−173, 2010.

41. Козак Р.О., Лобур Т.Б. Заходи щодо зменшення ризиків проникнення у безпровідний периметр корпоративної мережі // "Прогресивні напрямки розвитку технологічних комплексів ТК-2014", с. 127, 2014.

42. Кареліна О. Використання технології RFID в ERP-системах / О. Кареліна // Матеріали XXI наукової конференції ТНТУ ім. І. Пулюя, 16-17 травня 2019 року. — Т. : ТНТУ, 2019. — С. 55.

43. Кареліна О. Особливості використання технології RFID в інформаційному забезпеченні промислових підприємств // Соціально-економічні проблеми і держава. — 2019. — Вип. 1 (20). — С. 46-51

44. Системи протипожежного захисту: ДБН В.2.5-56:2014 – [Чинні від 2015-07-01]. – Державна служба надзвичайних ситуацій України, 2014 – 134с.

45. Системи пожежної та охоронної сигналізації / [Христич В.В., Дерев'янко О. А., Бондаренко С. М., Антошкін О. А.]. – Академія пожежної безпеки України, Харків, 2017. – 87с.

46. Наказ Про затвердження Правил пожежної безпеки в Україні: редакція від 3 жовтня 2017р. / Міністерство внутрішніх справ України, 2014. – 53с. (Нормативні директивні правові документи).

47. Наказ Про затвердження Примірної інструкції з охорони праці під час експлуатації електронно-обчислювальних машин: поточна редакція від 05.09.2013р. / Міністерство доходів і зборів України, 2013. – 7с. (Нормативні директивні правові документи).

48. Основні санітарні правила забезпечення радіаційної безпеки України. Наказ Міністерства охорони здоров'я України від 2 лютого 2005 року N 54, Стеблюк М.І. Цивільна оборона та цивільний захист: Підручник. – К.: Знання-Прес, 2007. – 487 с.

49. Васійчук В.О., Гончарук В.Є.та ін. Основи цивільного захисту. Навч. посібник / В.О. Васійчук, В.Є Гончарук, С.І. Качан, С.М. Мохняк. - Львів: Видавництво НУ "ЛП", 2010. - 417с.

50. Сакевич В.Ф., Томчук М.А. Основи розробки питань цивільної оборони в дипломних проектах, ВНТУ, 2008. - 141 с.

ДОДАТКИ

Додаток А – Скетч програми

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

#include <MFRC522.h>
#include <EEPROM.h>
MFRC522 _mfrc522(10, 9);
byte _mfrc522_notConnectCounter = 0;
bool _mfrc522_ConnectInpulse = 0;
bool _mfrc522_hasNewCard = 0;
bool _mfrc522_CardInfoOldState = 0;
byte _FLPArray116206301[11];
bool GSM_In_sms;
String GSM_Phone;
String GSM_text;
bool GSM_Reject_call;
bool GSM_start;
bool GSM_Net;
int GSM_dBm;
bool GSM_Received_SMS;
String GSM_Text_sms;
String GSM_T_Nomer;
bool GSM_Sent_SMS;
bool GSM_Call;
int GSM_Number_calls;
int GSM_Error_Code;
bool GSM_start_N; //Модем прислав повідомлення про те, що він стартував
bool GSM_final; // завершено обробку запиту
bool GSM_wait; // очікування відповіді на команду
bool GSM_past_SMS;
bool GSM_command_SMS;
bool GSM_past_Reject_call;
bool GSM_command_Reject_call;
//bool GSM_Call_P1;
char GSM_ch;            //символ отриманий в ком порт
byte GSM_errorCounter; // щотчик помилок
byte GSM_numberRepeats;
//byte GSM_Dial_response_code;
byte GSM_team_room; //номер поточної функції
byte GSM_count; // лічильник послідовності команд
unsigned long GSM_time; //час відправлення команди
unsigned long GSM_T_millis; //Час відправлення команди
unsigned long GSM_time_n; //час останнього запиту зв'язку з мережею
String GSM_line; // повідомлення від модуля
String GSM_reply; //відповідь модуля на команду
String GSM_Phone_T=""; //Телефон під час подачі команд на дзвінок чи відправку смс
SoftwareSerial Serial_S( 2 ,  3 );
int GSM_DTMF;
#define GSM_INCLUSION_DTMF
bool GSM_Outgoing_Call;
bool GSM_Take_Call;
bool GSM_Connect;
#define GSM_INCLUSION_CALLS

#define Module_Start "Call Ready"

#define GSM_TYPE_SIM800

struct RFID_MFRC522Struct
{
int freeCeilsCount;
int firstFreeCeilIndex;
bool isChange = 1;
};
RFID_MFRC522Struct RFID_MFRC522Struct_182368345;
struct RFID_MFRC522FindCellStructure
{byte statusFindValue;
int findIndex;
bool findPresence;
};
RFID_MFRC522FindCellStructure RFID_MFRC522FindCell_2_result;
String _gtv1;
```

```
bool _gtv2 = 0;
String _gtv3;
int _gtv4;
bool _gtv5 = 0;
float _gtv7;
bool _gtv8 = 0;
bool _gtv9 = 0;
bool _gtv10 = 0;
bool _gtv11 = 0;
bool _gtv12;
bool _gtv13 = 0;
bool _trgrt13 = 0;
bool _trgrt13I = 0;
bool _tim6I = 0;
bool _tim6O = 0;
unsigned long _tim6P = 0UL;
bool _trgrt5 = 0;
bool _trgrt5I = 0;
bool _bounse1S = 0;
bool _bounse1O = 0;
unsigned long _bounse1P = 0UL;
bool _trgrt11 = 0;
bool _trgrt11I = 0;
bool _trgrt9 = 0;
bool _trgrt9I = 0;
bool _trgrt12 = 0;
bool _trgrt12I = 0;
bool _trgrt6 = 0;
bool _trgrt6I = 0;
bool _trgrt16 = 0;
bool _trgrt16I = 0;
bool _trgrt17 = 0;
bool _trgrt17I = 0;
bool _trgrt2 = 0;
bool _trgrt2I = 0;
bool _trgrt15 = 0;
bool _trgrt15I = 0;
bool RFID_MFRC522SaveToStore_2_OldState;
bool _pzs2OES = 0;
bool _tim5I = 0;
bool _tim5O = 0;
unsigned long _tim5P = 0UL;
bool _tim11I = 0;
bool _tim11O = 0;
unsigned long _tim11P = 0UL;
bool _bounseInputD1S = 0;
bool _bounseInputD1O = 0;
unsigned long _bounseInputD1P = 0UL;
bool _tim9I = 0;
bool _tim9O = 0;
unsigned long _tim9P = 0UL;
bool _trgrt14 = 0;
bool _trgrt14I = 0;
bool _trgrt19 = 0;
bool _trgrt19I = 0;
bool _trgrt8 = 0;
bool _trgrt8I = 0;
bool _trgr2 = 0;
bool _trgrt20 = 0;
bool _trgrt20I = 0;
bool _bounseInputD3S = 0;
bool _bounseInputD3O = 0;
unsigned long _bounseInputD3P = 0UL;
bool _tim8I = 0;
bool _tim8O = 0;
unsigned long _tim8P = 0UL;
bool _trgrt22 = 0;
bool _trgrt22I = 0;
bool _trgrt3 = 0;
bool _trgrt3I = 0;
bool _tim3I = 0;
bool _tim3O = 0;
unsigned long _tim3P = 0UL;
bool _trgrt21 = 0;
bool _trgrt21I = 0;
bool _trgrt4 = 0;
bool _trgrt4I = 0;
bool _tim4I = 0;
bool _tim4O = 0;
unsigned long _tim4P = 0UL;
```

```
bool _tim7I = 0;
bool _tim7O = 0;
unsigned long _tim7P = 0UL;
bool RFID_MFRC522FindCell_2_OldState;
bool _tim1I = 0;
bool _tim1O = 0;
unsigned long _tim1P = 0UL;
bool _bounseInputD2S = 0;
bool _bounseInputD2O = 0;
unsigned long _bounseInputD2P = 0UL;
bool RFID_MFRC522ClearStore_2_OldState;
bool _trgr1 = 0;
bool _trgrt1 = 0;
bool _trgrt1I = 0;
bool _trgrt18 = 0;
bool _trgrt18I = 0;
bool _gen2I = 0;
bool _gen2O = 0;
unsigned long _gen2P = 0UL;
bool _trgrt10 = 0;
bool _trgrt10I = 0;
bool _trgs1 = 0;
bool _trgrt23 = 0;
bool _trgrt23I = 0;
bool _tim2I = 0;
bool _tim2O = 0;
unsigned long _tim2P = 0UL;
String _swi2;

void setup()
{
pinMode(1, INPUT_PULLUP);
pinMode(2, INPUT);
pinMode(3, INPUT_PULLUP);
pinMode(5, OUTPUT);
digitalWrite(5, 0);
pinMode(4, OUTPUT);
digitalWrite(4, 0);
pinMode(8, OUTPUT);
digitalWrite(8, 0);
pinMode(7, OUTPUT);
digitalWrite(7, 0);
pinMode(5, OUTPUT);
digitalWrite(5, 0);
pinMode(6, OUTPUT);
digitalWrite(6, 0);
pinMode(1, OUTPUT);
digitalWrite(1, 0);


_bounseInputD3O =  digitalRead(3);
_bounseInputD1O =  digitalRead(1);
_bounseInputD2O =  digitalRead(2);
SPI.begin();
pinMode(10, OUTPUT);
_mfrc522.PCD_Init();
RFID_MFRC522FindCell_2_result.statusFindValue = -1;
RFID_MFRC522FindCell_2_result.findIndex = -1;
RFID_MFRC522FindCell_2_result.findPresence = 0;
Serial_S.begin( 9600 );

 Serial_S.println("AT+IPR=9600");
Serial_S.println("AT&W");
}
void loop()
{

int _tempVariable_int;
_mfrc522_ConnectInpulse = _mfrc522.PICC_IsNewCardPresent();
if(_mfrc522_ConnectInpulse) {_mfrc522_ConnectInpulse =
_mfrc522.PICC_ReadCardSerial();}
if(_mfrc522_ConnectInpulse) {
_mfrc522_notConnectCounter = 0;
_mfrc522_hasNewCard = 1;
} else {
if(_mfrc522_notConnectCounter > 0) {
_mfrc522_hasNewCard = 0;
} else {_mfrc522_notConnectCounter =
_mfrc522_notConnectCounter + 1;}
_mfrc522.PICC_HaltA();
_mfrc522.PCD_StopCrypto1();}
```

```
if (_mfrc522_hasNewCard) {

if ( !_mfrc522_CardInfoOldState) {

_mfrc522_CardInfoOldState = 1;

_FLPArray116206301[10] = _mfrc522.uid.size;

 for(byte i = 0; i <=_FLPArray116206301[10] - 1; i++)
{_FLPArray116206301[i] = _mfrc522.uid.uidByte[i];}

}} else {

_mfrc522_CardInfoOldState = 0;

 for(byte i = 0; i <=11 ; i++) {_FLPArray116206301[i] = 0;}

}


bool  _bounceInputTmpD1 =  (digitalRead (1));


if (_bounseInputD1S)

   {

    if (millis() >= (_bounseInputD1P + 40))

      {_bounseInputD1O= _bounceInputTmpD1;
_bounseInputD1S=0;}

    }

else

   {

    if (_bounceInputTmpD1 != _bounseInputD1O )

      {_bounseInputD1S=1; _bounseInputD1P = millis();}

     }

bool  _bounceInputTmpD2 =  (digitalRead (2));


if (_bounseInputD2S)

   {

    if (millis() >= (_bounseInputD2P + 40))

      {_bounseInputD2O= _bounceInputTmpD2;
_bounseInputD2S=0;}

    }

else

   {

    if (_bounceInputTmpD2 != _bounseInputD2O )

      {_bounseInputD2S=1; _bounseInputD2P = millis();}

     }

bool  _bounceInputTmpD3 =  (digitalRead (3));


if (_bounseInputD3S)

   {

    if (millis() >= (_bounseInputD3P + 40))

      {_bounseInputD3O= _bounceInputTmpD3;
_bounseInputD3S=0;}

    }

else

   {

    if (_bounceInputTmpD3 != _bounseInputD3O )

      {_bounseInputD3S=1; _bounseInputD3P = millis();}

     }


//Плата:1

//назва:SIM800L

if (!(0)) { if (_trgrt16I) { _trgrt16 = 0;} else {_trgrt16 = 1; _trgrt16I =
1;} } else {_trgrt16 = 0; _trgrt16I = 0;};

 digitalWrite(5, _trgrt16);


GSM_In_sms = _gtv2;

GSM_Phone = String("+380...");

GSM_text = _gtv1;

GSM_Reject_call = _gtv8;

GSM_Received_SMS =0; //скидання статусу смс прийнято

GSM_Sent_SMS =0; //скидання статусу смс відправлено


if ( 1 ){

        if ( GSM_team_room==3 ){

                GSM_Number_calls =0;

                GSM_Text_sms="";

                GSM_T_Nomer ="";

        }

}


GSM_T_millis =millis();


                while(Serial_S.available()) //отримуємо дані з
модема

                        {GSM_ch = Serial_S.read();


                        if(GSM_ch == '\r') continue;

                        if(GSM_ch == '\n') {
GSM_GotLineFromNeoway(); GSM_line = ""; }

                        else  GSM_line += GSM_ch;}
```

```
if (GSM_wait)//очікування відповіді на команду
        {
        if(GSM_line.length()>0){
                if(GSM_line==F("> ")) {GSM_count++;
GSM_wait=0;GSM_line = "";} }
        if(GSM_T_millis>GSM_time)       {GSM_err (2,0);}
        }

if (GSM_Call&&(GSM_team_room!=1)  )
        {
        GSM_team_room=1;
        GSM_Number_calls=1; //скидання числа гудків
        // GSM_resetExpectations() ;
         }

if (GSM_start_N){if (!GSM_wait){if ( GSM_setup_comand()
){GSM_start_N=0; GSM_start =1; }}}// ініціалізація

if (GSM_start)//не працює без команди старт
        {
//_____Провірка статуса мережі
                if ((30000 >0)&&(GSM_team_room==0))
                        {
                        if((GSM_T_millis-GSM_time_n) >
30000){ GSM_team_room=3;}
                        }
//_____кінець перевірки статусу мережі

//_____відправка СМС
        if (  GSM_In_sms  && !GSM_past_SMS  ) {  if (
GSM_number_search()) GSM_command_SMS =1;       }              //
поступленняе команди на відправу СМС. перевірка коректності
номера
        GSM_past_SMS= GSM_In_sms;
        if ( GSM_command_SMS && (GSM_team_room==0) )
{GSM_team_room=2;  GSM_command_SMS =0;}  // почати
відправлення СМС
//_____кінець відправлення СМС

//_____Команда скидання виклику чи розрив розмови
        if (  GSM_Reject_call  && !GSM_past_Reject_call  ) {
GSM_command_Reject_call =1; }
        GSM_past_Reject_call=GSM_Reject_call;
//_____Конец кінець команди

#ifdef GSM_INCLUSION_USSD // обробка виклику USSD запитів
        GSM_causeProcessingUSSD();
#endif

#ifdef GSM_INCLUSION_DATA_TIME //обробка часу, дати,
оператора
        GSM_causeProcessingDataTime();
#endif

#ifdef GSM_INCLUSION_CALLS // вихідний виклик
        if (GSM_causeProcessingCall()) {
                if( GSM_number_search()){
                        GSM_wait=1; // чекати
                        GSM_time =millis() + 20000; //час
очікування
                        Serial_S.println( "ATD"+
GSM_Phone_T+";" );
                        GSM_team_room=5;
        #ifdef GSM_DEBUG // отладка
                Serial.println("Command: ATD"+
GSM_Phone_T+";" );
        #endif
                }
        }
#endif
//________ обробка різних процесів
        switch ( GSM_team_room ) {
        case 0:
                        break;
        case 1: //обробка вхідних викликів
                if (!GSM_wait){
                        if ( GSM_command_Reject_call ) {
                                if (
GSM_singleCommand(F("ATH0"), 10) )
{GSM_command_Reject_call=0; GSM_Call=0; GSM_count=0;
GSM_final=1;}
                                }
#ifdef GSM_INCLUSION_CALLS
#ifndef GSM_TYPE_M590
                if (GSM_causeProcessingTake()){
GSM_Connect=1; GSM_team_room=5;}
```

```
#endif

#endif

                        }

                        if (!GSM_Call){ GSM_final =1;  GSM_T_Nomer
="";}

                        break;

                case 2:

                        if (!GSM_wait){  GSM_final=GSM_sms_sent();}
// відправка повідомлення

                        break;

                case 3:

                        if (!GSM_wait){
GSM_final=GSM_net_status();}// перевірка зв'язку з мережкю

                        break;

                case 4:     //приход смс

                        //if (!GSM_wait){ GSM_final=1; GSM_count=0;
GSM_Error_Code =0;   }//

                        break;

#ifdef GSM_INCLUSION_CALLS

                case 5:     //розмова

                        if (!GSM_wait) {

                                if ( GSM_command_Reject_call ) {

                                        if (
GSM_singleCommand(F("ATH0"), 10))
{GSM_command_Reject_call=0; GSM_Call=0;  GSM_count=0;
GSM_Connect=0; GSM_final=1;}

                                }

                                GSM_count=0;


                        }

                        break;

#endif

#ifdef GSM_INCLUSION_DATA_TIME

                case 6:     // дата час

                        if (!GSM_wait){if ( GSM_DataTime() ){
GSM_final =1;}}// запит балансу

                        break;

#endif

#ifdef GSM_INCLUSION_USSD

                case 7:     // USSD

                        if (!GSM_wait){ GSM_final=GSM_USSD();}//
запит балансу

                        break;

#endif
```

```
                }

//_________ Кінець обробки різних процесів


                if (GSM_final){GSM_team_room=0; GSM_final=0;} //має
бути останнім

}


_gtv3 = GSM_T_Nomer;

digitalWrite(7, GSM_Net);

_gtv5 = ((_gtv3).equals(String("+38XXXXXXXXX")));

GSM_Outgoing_Call = 0;

GSM_Take_Call = ((_gtv3).equals(String("+38XXXXXXXX")));




_gtv4 = GSM_DTMF;


//Плата:2

//назва:Вольтметр / відправка SMS

if (!(0)) { if (! _gen2I) { _gen2I = 1; _gen2O = 1; _gen2P = millis(); } }
else { _gen2I = 0 ; _gen2O= 0;}

 if (_gen2I) {  if ( _isTimer ( _gen2P , 500 )) { _gen2P = millis();
_gen2O = ! _gen2O;}}

_gtv10 = _gen2O;

if (_gen2O) {

_gtv7 = ((map(( (analogRead (0))), (0), (1023), (0),
(4700))))/(1000.00);

}

if (_gtv11) { if (_trgrt4I) { _trgrt4 = 0;} else {_trgrt4 = 1; _trgrt4I = 1;} }
else {_trgrt4 = 0; _trgrt4I = 0;};

 if(_trgrt4) {_tim11O = 1; _tim11I = 1;} else { if(_tim11I) {_tim11I = 0;
_tim11P = millis();} else { if (_tim11O) {if ( _isTimer(_tim11P, 1000))
_tim11O = 0;}}}

if(_tim11O)

{_swi2=String("Pronuknennya");}

else

{_swi2=((String("Voltage ")) + (( _floatToStringWitRaz(_gtv7,2))) +
(String(" V")));}

_gtv1 = _swi2;

if ((_gtv4) == (9)) { if (_trgrt1I) { _trgrt1 = 0;} else {_trgrt1 = 1;
_trgrt1I = 1;} } else {_trgrt1 = 0; _trgrt1I = 0;};

 if (!(_trgrt1))

{ if (_tim7I) { if (_isTimer(_tim7P, 3000)) {_tim7O = 1;}} else {_tim7I
=1; _tim7P = millis();}} else {_tim7O = 0; _tim7I = 0;}
```

```
if (_tim7O) { if (_trgrt17I) { _trgrt17 = 0;} else {_trgrt17 = 1; _trgrt17I = 1;} } else {_trgrt17 = 0; _trgrt17I = 0;};

 if ((_gtv7) < (2.7)) { if (_trgrt2I) { _trgrt2 = 0;} else {_trgrt2 = 1; _trgrt2I = 1;} } else {_trgrt2 = 0; _trgrt2I = 0;};

 if ((_gtv7) < (3.5)) { if (_trgrt5I) { _trgrt5 = 0;} else {_trgrt5 = 1; _trgrt5I = 1;} } else {_trgrt5 = 0; _trgrt5I = 0;};

 if ((_gtv7) < (4.8)) { if (_trgrt18I) { _trgrt18 = 0;} else {_trgrt18 = 1; _trgrt18I = 1;} } else {_trgrt18 = 0; _trgrt18I = 0;};

 if (_gtv11) { if (_trgrt3I) { _trgrt3 = 0;} else {_trgrt3 = 1; _trgrt3I = 1;} } else {_trgrt3 = 0; _trgrt3I = 0;};

 _gtv2 = ( (( (_trgrt17) || (_trgrt18) || (_trgrt5) || (_trgrt2) || (_trgrt3) )) && (_gtv9) );

_gtv8 = (_gtv4) == (9);


//Плата:3

//назва:Сигналізація ON/Off

bool   _bounceTmpD1 = !(_bounseInputD1O);


if (_bounse1S)

   {

    if (millis() >= (_bounse1P + 40))

       {_bounse1O=  _bounceTmpD1; _bounse1S=0;}

    }

else

   {

    if ( _bounceTmpD1 != _bounse1O )

       {_bounse1S=1; _bounse1P = millis();}

     }

if(_bounse1O) _trgr1 = 0;

if(( (_bounseInputD1O) && (_gtv13) )) _trgr1 = 1;

if (( (_trgr1) && (!(_bounseInputD1O)) )) { if (_trgrt15I) { _trgrt15 = 0;} else {_trgrt15 = 1; _trgrt15I = 1;} } else {_trgrt15 = 0; _trgrt15I = 0;};

 if ((_gtv4) == (2)) { if (_trgrt22I) { _trgrt22 = 0;} else {_trgrt22 = 1; _trgrt22I = 1;} } else {_trgrt22 = 0; _trgrt22I = 0;};

 if (( (_gtv13) && (!(_bounseInputD1O)) )) { if (_trgrt19I) { _trgrt19 = 0;} else {_trgrt19 = 1; _trgrt19I = 1;} } else {_trgrt19 = 0; _trgrt19I = 0;};

 if ((_gtv4) == (1)) { if (_trgrt23I) { _trgrt23 = 0;} else {_trgrt23 = 1; _trgrt23I = 1;} } else {_trgrt23 = 0; _trgrt23I = 0;};

 if(( (_trgrt19) || (_trgrt22) )) _trgr2 = 0;

if(( (_trgrt15) || (_trgrt23) )) _trgr2 = 1;

if(( (_gtv10) && (_trgr2) )) {_tim8O = 1; _tim8I = 1;} else { if(_tim8I) {_tim8I = 0; _tim8P = millis();} else { if (_tim8O) {if ( _isTimer(_tim8P, 50)) _tim8O = 0;}}}

digitalWrite(6, ( (_trgr1) || (_tim8O) ));

_gtv9 = _trgr2;

if ((_gtv4) == (2)) { if (_trgrt11I) { _trgrt11 = 0;} else {_trgrt11 = 1; _trgrt11I = 1;} } else {_trgrt11 = 0; _trgrt11I = 0;};

 if(_trgrt11) {_tim1O = 1; _tim1I = 1;} else { if(_tim1I) {_tim1I = 0; _tim1P = millis();} else { if (_tim1O) {if ( _isTimer(_tim1P, 3000)) _tim1O = 0;}}}

if ((_gtv4) == (1)) { if (_trgrt10I) { _trgrt10 = 0;} else {_trgrt10 = 1; _trgrt10I = 1;} } else {_trgrt10 = 0; _trgrt10I = 0;};

 if(_trgrt10) {_tim9O = 1; _tim9I = 1;} else { if(_tim9I) {_tim9I = 0; _tim9P = millis();} else { if (_tim9O) {if ( _isTimer(_tim9P, 2000)) _tim9O = 0;}}}

if(( (_tim9O) || (( (_tim1O) && (_gtv10) )) ))

{if((!_pzs2OES)){ tone(4, 1000); _pzs2OES =1;}} else {if(_pzs2OES){noTone(4); _pzs2OES =0;  }}


//Плата:4

//назва:Работа сигналізації

if (_bounseInputD2O) { if (_trgrt12I) { _trgrt12 = 0;} else {_trgrt12 = 1; _trgrt12I = 1;} } else {_trgrt12 = 0; _trgrt12I = 0;};

 if(_trgrt12) _trgs1 = 1;

if(!(_gtv9)) _trgs1 = 0;

if (_trgs1) { if (_trgrt20I) { _trgrt20 = 0;} else {_trgrt20 = 1; _trgrt20I = 1;} } else {_trgrt20 = 0; _trgrt20I = 0;};

 if (_bounseInputD1O) { if (_trgrt21I) { _trgrt21 = 0;} else {_trgrt21 = 1; _trgrt21I = 1;} } else {_trgrt21 = 0; _trgrt21I = 0;};

 _gtv11 = ( (( (_trgrt20) || (_trgrt21) )) && (_gtv9) );

if(( (( (_trgrt20) || (_trgrt21) )) && (_gtv9) )) {_tim6O = 1; _tim6I = 1;} else { if(_tim6I) {_tim6I = 0; _tim6P = millis();} else { if (_tim6O) {if ( _isTimer(_tim6P, 10000)) _tim6O = 0;}}}

digitalWrite(1, ( (_gtv10) && (_tim6O) ));


//Плата:5

//назва:RFID

if (_mfrc522_hasNewCard) {

if ( !_mfrc522_CardInfoOldState) {

_mfrc522_CardInfoOldState = 1;

_FLPArray116206301[10] = _mfrc522.uid.size;

 for(byte i = 0; i <=_FLPArray116206301[10] - 1; i++) {_FLPArray116206301[i] = _mfrc522.uid.uidByte[i];}

}} else {

_mfrc522_CardInfoOldState = 0;

 for(byte i = 0; i <=11 ; i++) {_FLPArray116206301[i] = 0;}

}

if (!(_bounseInputD3O))

{ if (_tim2I) { if (_isTimer(_tim2P, 5000)) {_tim2O = 1;}} else {_tim2I =1; _tim2P = millis();}} else {_tim2O = 0; _tim2I = 0;}
```

```
if (_tim2O) { if (_trgrt9I) { _trgrt9 = 0;} else {_trgrt9 = 1; _trgrt9I = 1;}
} else {_trgrt9 = 0; _trgrt9I = 0;};

 if(_trgrt9) {_tim4O = 1; _tim4I = 1;} else { if(_tim4I) {_tim4I = 0;
_tim4P = millis();} else { if (_tim4O) {if ( _isTimer(_tim4P, 1000))
_tim4O = 0;}}}

digitalWrite(5, _mfrc522_hasNewCard);

if (_mfrc522_hasNewCard){

if( ! RFID_MFRC522FindCell_2_OldState){

RFID_MFRC522FindCell_2_OldState = 1;

_rfid_MFRC522FindCellsToEEPOM(0, 0x0 , _FLPArray116206301, 10,
0, &RFID_MFRC522FindCell_2_result);

}} else {RFID_MFRC522FindCell_2_OldState = 0;

RFID_MFRC522FindCell_2_result.statusFindValue = -1;

RFID_MFRC522FindCell_2_result.findIndex = -1;

RFID_MFRC522FindCell_2_result.findPresence = 0;}

if ((_gtv4) == (4)) { if (_trgrt6I) { _trgrt6 = 0;} else {_trgrt6 = 1;
_trgrt6I = 1;} } else {_trgrt6 = 0; _trgrt6I = 0;};

 if (( (RFID_MFRC522FindCell_2_result.findPresence) || (_trgrt6) )) {
if (_trgrt14I) { _trgrt14 = 0;} else {_trgrt14 = 1; _trgrt14I = 1;} } else
{_trgrt14 = 0; _trgrt14I = 0;};

 if(_trgrt14) {_tim5O = 1; _tim5I = 1;} else { if(_tim5I) {_tim5I = 0;
_tim5P = millis();} else { if (_tim5O) {if ( _isTimer(_tim5P, 3000))
_tim5O = 0;}}}

digitalWrite(8, !(( (_tim5O) || (_gtv12) )));

if (( (RFID_MFRC522FindCell_2_result.findPresence) || (_trgrt6) )) {
if (_trgrt13I) { _trgrt13 = 0;} else {_trgrt13 = 1; _trgrt13I = 1;} } else
{_trgrt13 = 0; _trgrt13I = 0;};

 _gtv13 = ( (RFID_MFRC522FindCell_2_result.findPresence) ||
(_trgrt6) );

if (( (_mfrc522_hasNewCard) && (!(_bounseInputD3O)) )) { if
(_trgrt8I) { _trgrt8 = 0;} else {_trgrt8 = 1; _trgrt8I = 1;} } else {_trgrt8
= 0; _trgrt8I = 0;};

  _rfid_MFRC522FreeCellsOnEEProm(0, 0x0, 10,
&RFID_MFRC522Struct_182368345);

if (_trgrt8){

if( ! RFID_MFRC522SaveToStore_2_OldState){

RFID_MFRC522SaveToStore_2_OldState = 1;

_tempVariable_int =
RFID_MFRC522Struct_182368345.firstFreeCeilIndex;

if((_tempVariable_int >= 0)&&(_tempVariable_int < 10)){

_tempVariable_int = ((_tempVariable_int)*12) + 0;

for(int i=0; i<11;i++) {updateByteToEEPROM((_tempVariable_int + i),
0, 0x0, _FLPArray116206301[i]);};

updateByteToEEPROM((_tempVariable_int + 11), 0, 0x0, 2);

RFID_MFRC522Struct_182368345.isChange = 1;

}

}} else {RFID_MFRC522SaveToStore_2_OldState = 0;}

if(( (_trgrt8) || (_trgrt13) )) {_tim3O = 1; _tim3I = 1;} else { if(_tim3I)
{_tim3I = 0; _tim3P = millis();} else { if (_tim3O) {if ( _isTimer(_tim3P,
100)) _tim3O = 0;}}}

digitalWrite(4, ( (_tim4O) || (_tim3O) ));

if (_trgrt9){

if( ! RFID_MFRC522ClearStore_2_OldState){

RFID_MFRC522ClearStore_2_OldState = 1;

for(int i=0; i < 10;i++) {updateByteToEEPROM(((i*12) +11), 0, 0x0,
0);}

RFID_MFRC522Struct_182368345.isChange = 1;

}} else {RFID_MFRC522ClearStore_2_OldState = 0;}

}

String  _floatToStringWitRaz(float value, int raz)

{   return String(value,raz);}bool _isTimer(unsigned long startTime,
unsigned long period )

 {  unsigned long currentTime;

currentTime = millis();

if (currentTime>= startTime) {return (currentTime>=(startTime +
period));} else {return (currentTime >=(4294967295-
startTime+period));}

 }

void GSM_GotLineFromNeoway()

{

bool flag_=0; //Аналіз телефонної книги

static bool isStringMessage;

#ifdef GSM_DEBUG

Serial.print("Message module: ");//Тест

Serial.println(GSM_line); //Тест

Serial.print("GSM_team_room: ");//Тест

Serial.println(GSM_team_room); //Тест

#endif


if (GSM_line.length()>0)

            {


//Обробка вхідних СМС

                        if (isStringMessage){

                                    GSM_Text_sms  =  GSM_line ;
//вивід тексту повідомлення

                                    GSM_Received_SMS  =true; //
імпульс отримання повідомлення

                                    isStringMessage=0;


                                    return;
```

```
                }
        if (GSM_line.startsWith("+CMT")) // прийшло СМС
                {
                flag_=1;
                GSM_resetExpectations();
                isStringMessage=1;
                GSM_reply="";
                }
//кінець обробки вхідних СМС

// Обробка вхідних викликів

        if (GSM_Call)
                {
                if (GSM_line.startsWith(F("+CLIP")))
{flag_=1;GSM_reply="";}
                else if (GSM_line == F("NO CARRIER")) {
GSM_Call=0; }
                }
        if (GSM_line == F("RING")){ //Вхідний дзвінок

        if(!GSM_Call)GSM_resetExpectations();
                                        GSM_Call=1;
                                        GSM_Number_calls ++;

                                        }
// ____ Обробка USSD sim800
#ifdef GSM_TYPE_SIM800
#ifdef GSM_INCLUSION_USSD
if (GSM_line.startsWith("+CUSD")) // прийшла відповідь по USSD
                {
                GSM_wait=0;
                GSM_count++;
                GSM_USSD_return  =
GSM_substring_commas(GSM_line,5) ;
                }
#endif
#endif
//_______Кінець обробки USSD SIM800

#ifdef GSM_INCLUSION_CALLS // якщо додано блок роботи з
дзвінками
#ifdef GSM_INCLUSION_DTMF // якщо додано блок DTMF
        if ( GSM_Connect ){    if
(GSM_line.startsWith(F("+DTMF"))) {GSM_ID_DTMF(GSM_line);} }
#endif
        if (GSM_team_room==5)
                {
                if (GSM_line ==
F("CONNECT"))GSM_Connect=1;
                else if (GSM_line == F("BUSY")) GSM_err
(505,2);
                else if (GSM_line == F("NO ANSWER"))
GSM_err (506,2);
                else if (GSM_line == F("NO CARRIER")) {
                                        if
(GSM_Connect){GSM_Connect=0; GSM_team_room=0;}
                                        else GSM_err (507,2);
                                }
                }
#endif
// кінець обробки вхідних викликів



        if(GSM_line == F("ERROR")) GSM_err (1, 0);
        else if(GSM_line == F("OK")){  GSM_wait=0;
GSM_count++; GSM_errorCounter=0; }
        else {
                GSM_reply+=GSM_line;
                }
        if (GSM_reply.length()>  160  ){GSM_reply=""; GSM_err
(10100,2);}//захист від переповнення рядка



if (GSM_line == Module_Start){ GSM_start_N=1;  GSM_start=0;
GSM_err (0 , 2);  }//повідомлення модуля про кінець загрузки


if (flag_) //отримання номера відправника
                {
                 GSM_T_Nomer =  GSM_substring_commas(
GSM_line, 4) ;
                if( GSM_T_Nomer.indexOf("+")==-1)
GSM_T_Nomer=(String( "+" ))+ GSM_T_Nomer;
#ifdef TELEFONNAYA_BOOK // якщо використовується телефонна
книга
```

```
                int  GSM_ind_arr= GSM_StringArray(
GSM_Telefon,  GSM_T_Nomer );

                if ( GSM_ind_arr>-1)  GSM_T_Nomer =
GSM_Contact [  GSM_ind_arr ];

#endif

                flag_=0;

                }

#ifdef TELEFONNAYA_BOOK //якщо використовується телефонна
книга

        if (!GSM_start){

                if ( GSM_line.startsWith("+CPBF")) //
прийшов запис телефонної книги

                        {if( GSM_index_array<
GSM_Number_Contacts ){

                                GSM_Telefon[
GSM_index_array ] =  GSM_substring_commas(  GSM_line, 5);

                                GSM_Contact[
GSM_index_array ]=  GSM_substring_commas(  GSM_line, (
GSM_line.lastIndexOf(","))) ;

                                 GSM_index_array ++;}

                        }

        }

#endif

}

// Отримали рядок відповіді від Neoway. Будь-які відповіді
приходять у вигляді

// або однієї, або декількох рядків, так що одиниця, якій ми
повинні

// оперувати - саме рядок

// F ("") -зберігати рядок в Флеш пам'яті

}

void GSM_AT_Command(const __FlashStringHelper* str, String str2,
byte _numRep)

{

 GSM_numberRepeats=_numRep;

 GSM_reply="";// скидання вмісту попередньої відповіді модуля

 GSM_wait=1;

 GSM_time =millis() + 20000; // час очікування

Serial_S.write("AT+");

Serial_S.print(str);

Serial_S.print(str2);

Serial_S.write("\"\r\n");

#ifdef GSM_DEBUG // отладка

        Serial.write("Command: AT+");

        Serial.print(str);

        Serial.print(str2);

        Serial.write("\"\r\n");

#endif

}

bool GSM_net_status()

{

switch (  GSM_count ) {

case 0:

         GSM_count ++;

        break;

case 1:

         GSM_AT_Command(F("CREG?") , 5);          //перевірка
реєстрації в мережі

        break;

case 2:

        if ( GSM_analysisResponse(F("+CREG") )){

                if( ( GSM_reply.endsWith("1")) || (
GSM_reply.endsWith("5"))) { GSM_Net =1;} // 1-В домашній
мережі, 5- в роумінгу

                else { GSM_Net =0;}

                }

        break;

case 3: //перевірка рівня сигналу

         GSM_AT_Command(F("CSQ") , 5 ) ;

        break;

case 4:

        if ( GSM_analysisResponse(F("+CSQ") )){
                //

                int t1= GSM_reply.indexOf(": ");

                if(t1>-1)    GSM_dBm  =(( ( (
GSM_reply.substring(t1+2,t1+4) ).toInt())*(2) )-(113));


                if(  GSM_dBm >0)   GSM_dBm =-120; //нема
мережі

                }

        break;

case 5:

         GSM_count=0;

         GSM_time_n=millis();// час останнього запиту

         GSM_Error_Code =0; //скидання помилки

        return 1;

        break;
```

```
}
return 0;
}
bool GSM_sms_sent()
{
switch (  GSM_count ) {
case 0:
                GSM_count ++;
                break;
case 1:
                GSM_AT_Command(F("CMGF=1"),1);                          //
преведення повідомлень у текстовий режим
                break;
case 2:
                GSM_AT_Command(F("CMGS=\""),  GSM_Phone_T ,1 );
// відправка команди на початковий етап відправки СМС
                break;
case 3: // передача тексту СМС та відправлення повідомлення
                if (  GSM_errorCounter ){  GSM_err( (200+ GSM_count) ,
2);  break; }//вихід з відправлень команд, якщо досягнуто ліміт
помилок
                Serial_S.println( GSM_text +String( (char)26 ));

                GSM_wait=1;
                GSM_time =millis() + 20000; // час очікування
#ifdef GSM_DEBUG // отладка
                Serial.println( GSM_text +String( (char)26 ));

#endif
                break;
case 4:
                GSM_count=0;
                GSM_Sent_SMS =1;
                GSM_Error_Code =0; //скидання помилки
                return 1;
                break;
}
return 0;
}
String GSM_substring_commas(String text_analyze, int position)
{
int t1=text_analyze.indexOf("\"", position);
int t2=text_analyze.indexOf("\"",(t1+1));
if(t1>-1){
                if(t2>-1)  return text_analyze.substring(t1+1,t2);
                else return text_analyze.substring(t1+1);}
return "";
}
bool GSM_number_search()
{
if ( GSM_Phone.length() <4) {GSM_Error_Code =10005; return 0;}
GSM_Phone_T = GSM_Phone ;

#ifdef TELEFONNAYA_BOOK
int ind_arr;
ind_arr= GSM_StringArray( GSM_Contact ,  GSM_Phone  );
if (ind_arr>-1)  GSM_Phone_T = GSM_Telefon [ ind_arr ];
else if(GSM_White_list ) {GSM_Error_Code =10006; return 0;}
#endif

return 1;
}
void GSM_resetExpectations()
{
//скидання поточної операції
GSM_wait=0; //скидання очікування відповіді
GSM_count=0; //скидання стану обробки інших функцій
GSM_time_n=GSM_T_millis;//відстрочка перевірки статусу
мережі
GSM_team_room=0;

}
void GSM_err(int error, byte _ret)
{
GSM_errorCounter++ ;
GSM_reply=""; //скидання вмісту попередньої відповіді модуля
GSM_wait =0;
if (_ret==0){
                if ( GSM_errorCounter >GSM_numberRepeats){
                                error = GSM_team_room*100+ GSM_count;
                                _ret=2;}
                else return;
```

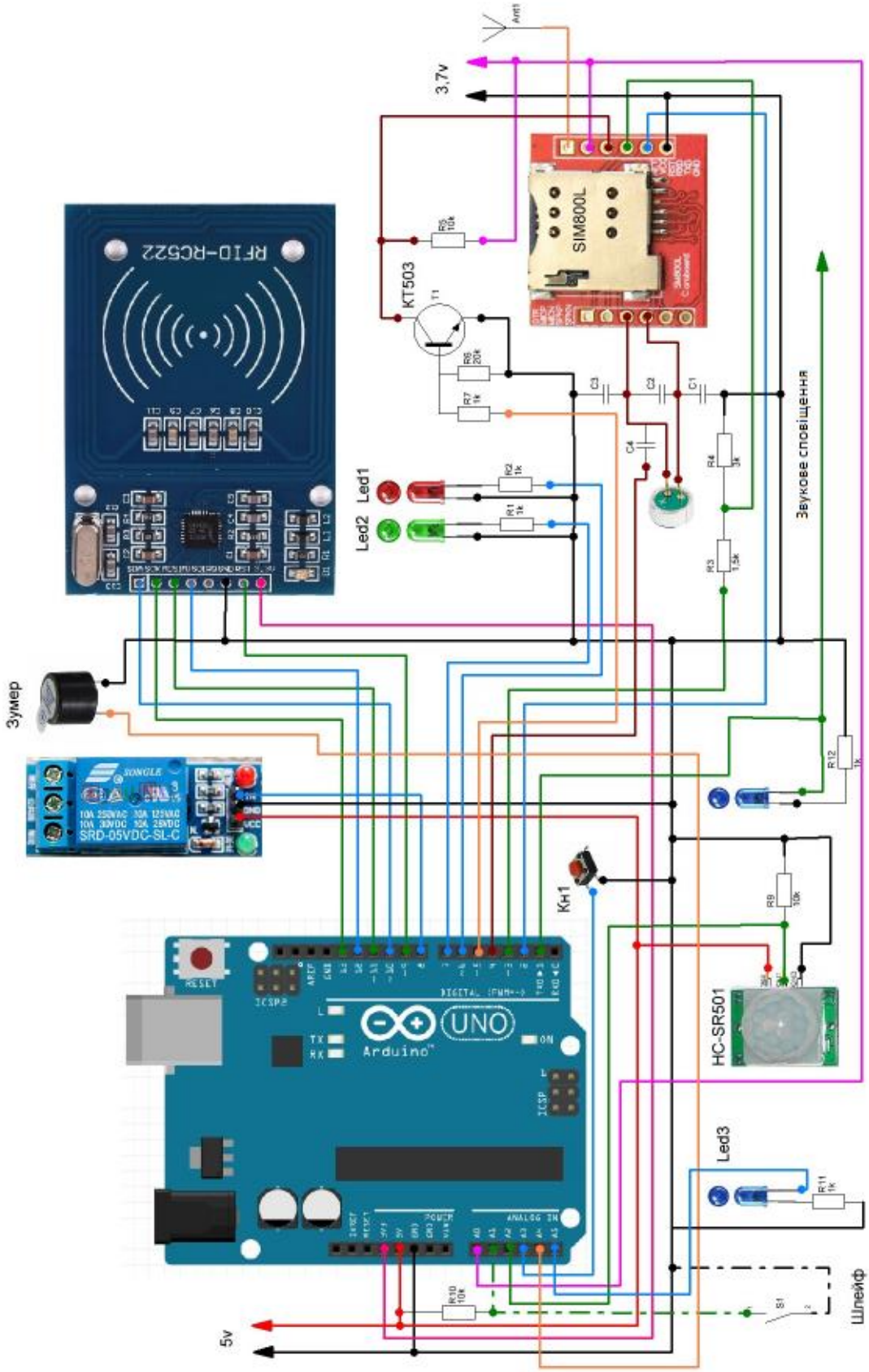
```
}
if ( _ret==1)  GSM_count--;
else if (_ret==2 ){
        GSM_errorCounter=0;
        GSM_Error_Code =error;
        if (GSM_team_room==3){GSM_Net =0;GSM_dBm =-120;}
        GSM_resetExpectations();
}
}
void GSM_AT_Command(const __FlashStringHelper* str, byte
_numRep)
{
#ifdef GSM_DEBUG // отладка
        Serial.write("AT+ _ ");
#endif
Serial_S.write("AT+");
 GSM_Command( str,_numRep  ) ;
}
bool GSM_analysisResponse(const __FlashStringHelper* response)
{
#ifdef GSM_DEBUG
Serial.print("Expected Answer: ");//Тест
Serial.println(response); //Тест
Serial.print("Real  Answer: ");//Тест
Serial.println(GSM_reply); //Тест
#endif

if (GSM_reply.startsWith(response)){GSM_count++; return 1; }
GSM_err(0,1);
return 0;
}
bool GSM_singleCommand(const __FlashStringHelper* str, byte
_numRep)
{
switch ( GSM_count ) {
case 0:
        GSM_Command(str, _numRep) ;
        break;
case 1:
        GSM_count=0;
        GSM_Error_Code =0; //скидання помилки
        return 1;
        break;
}
return 0;
}
void GSM_Command(const __FlashStringHelper* str, byte
_numRep)
{
GSM_reply=""; //скидання вмісту попередньої відповіді модуля
GSM_numberRepeats=_numRep;
GSM_wait=1; // чекати
GSM_time =millis() + 20000; // час очікування
Serial_S.println(str);
#ifdef GSM_DEBUG // отладка
        Serial.write("Command: ");
        Serial.println(str);
#endif

}
void GSM_ID_DTMF(String st)
{
int www=(int(st.charAt(7)));
switch ( www ) {
case 35:
        GSM_DTMF =14;
        break;
case 42:
        GSM_DTMF =15;
        break;
case 65:
        GSM_DTMF =10;
        break;
case 66:
        GSM_DTMF =11;
        break;
case 67:
        GSM_DTMF =12;
        break;
case 68:
        GSM_DTMF =13;
```

```
            break;
 default:
if (www>47 && www<58) GSM_DTMF =www-48;
}

}
bool GSM_causeProcessingCall()
{
static bool past_Outgoing_Call=0;
static bool command_Outgoing_Call=0;

if (  GSM_Outgoing_Call   && !past_Outgoing_Call   )
command_Outgoing_Call  =1;
past_Outgoing_Call  = GSM_Outgoing_Call ;
if (  command_Outgoing_Call  && (GSM_team_room==0) )          {
             command_Outgoing_Call  =0;  return 1;} //почати запит
часу
return 0;
}
bool GSM_causeProcessingTake()
{
static bool past_Take_Call;
static bool command_Take_Call;

if (   GSM_Take_Call    && !past_Take_Call    ) {
command_Take_Call=1;  }
            past_Take_Call   =  GSM_Take_Call;
if (  command_Take_Call    ) {
            switch (  GSM_count ) {
            case 0:
                        GSM_count++;
                        break;
            case 1:
                        GSM_Command(F("ATA"), 5);     // "підняти
трубку"
            #ifdef GSM_DEBUG // отладка
                        Serial.println("Command: ATA" );
            #endif
                        break;
            case 2:
                        GSM_count=0;
                        command_Take_Call  =0;
                        return 1;
                        break;
            }
}
return 0;
}
bool GSM_setup_comand()
{
switch (  GSM_count ) {
case 0:
             GSM_count ++;
            break;
case 1:
             GSM_Command(F("ATE0") ,2 ) ; //відключення exo
відповіді
            break;
case 2:
             GSM_AT_Command(F("CPAS"),4);
            // перевірка готовності модуля до роботи
            break;
case 3:
             GSM_analysisResponse(F("+CPAS: 0") );
            break;
case 4:
             GSM_AT_Command(F("CREG?"),10);          //перевірка
реєстрації в мережі
            break;
case 5:
            if ( GSM_analysisResponse(F("+CREG") )){
                        if( ( GSM_reply.endsWith("1")) || (
GSM_reply.endsWith("5"))) { GSM_Net =1;} // 1-В домашній
мережі, 5- в роумінгу
                        else { GSM_err(0,1);  GSM_count--; }

                        }
            break;
case 6:
            GSM_AT_Command(F("CSCB=1") ,2 ) ;                       //
Відключення прийому широкомовних повідомлень
            break;
case 7:
            GSM_AT_Command(F("CLIP=1") ,2 ) ;                       //
настройки автоматичного визначення номера
```

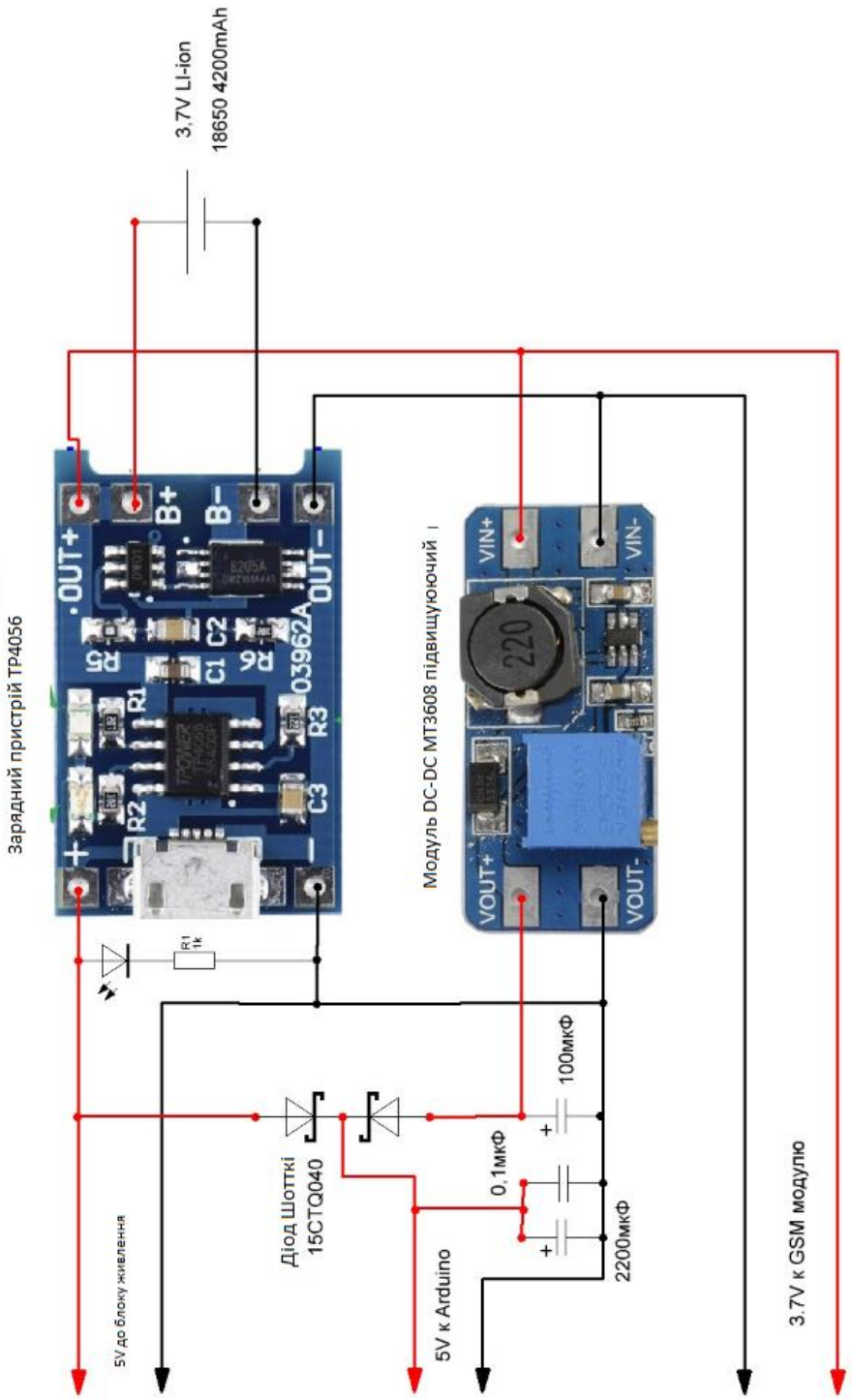
```
            break;
case 8:
            GSM_AT_Command(F("CMGF=1") ,2 ) ; //настройки формата SMS повідомлень, текстовий
            break;
case 9:
            GSM_AT_Command(F("CSCS=\"GSM\"") ,2 ) ;          // вибір кодування тексту, кодування ASCII
            break;
case 10:
            GSM_AT_Command(F("CNMI=2,2,0,0,0") ,2 ) ; //видавати повідомлення в термінал без збереження в пам'ять
            break;
case 11:
#ifdef GSM_INCLUSION_DTMF // якщо додано блок DTMF
            GSM_AT_Command(F("DDET=1") ,5 ); // включення роботи з DTMF
#else
            GSM_count ++;
#endif
            break;
#ifdef TELEFONNAYA_BOOK // якщо використовується телефоннв книга
case 12:
            GSM_AT_Command(F("CPBS?"),2);

            break;
case 13:
            GSM_analysisResponse(F("+CPBS: \"SM\"") );

            break;
case 14:
            GSM_AT_Command(F("CPBF=\""), GSM_Name_Contact , 2 ) ;
            GSM_index_array =0;
            break;
case 15:
#else //якщо не використовуємо телеф. кн.
case 12:
#endif //кінець вибору в тел.кн.
            GSM_count=0;
            GSM_Error_Code =0; //скидання помилки
            return 1;
            break;
}
return 0;
}
void  _rfid_MFRC522FreeCellsOnEEProm(int startAddres, byte chipAddres, int storeSize, struct RFID_MFRC522Struct *storeSruct)
{
 if (! storeSruct->isChange) {return;};
storeSruct->freeCeilsCount = 0;
storeSruct->firstFreeCeilIndex = -1;
for(int i=0; i<storeSize; i++) {if (( readByteFromEEPROM((startAddres + (i * 12) + 11), 0,chipAddres)) == 0){
storeSruct->freeCeilsCount = storeSruct->freeCeilsCount +1;
if(storeSruct->firstFreeCeilIndex == -1){storeSruct->firstFreeCeilIndex = i;}
}
}
storeSruct->isChange=0;
}
void  _rfid_MFRC522FindCellsToEEPOM(int startAddes, byte chipAddres, byte *keyArray, int storeSize, bool findFromBlocked, struct RFID_MFRC522FindCellStructure *storeSruct)
{
storeSruct->statusFindValue = -1;
storeSruct->findIndex = -1;
storeSruct->findPresence = 0;
byte tempValue;
byte status;
bool isFind;
for(int i = 0; i < storeSize; i++)
{
 status = readByteFromEEPROM(((i * 12) + 11), 0, chipAddres);
 if (!(status == 0 )) {
   if (!((status == 1 ) && ((! findFromBlocked)) ) ) {
tempValue =  readByteFromEEPROM((startAddes+(i*12)), 0, chipAddres);
if(tempValue == keyArray[0])
 {
isFind=1;
for (int id = 1; id < 11; id++)
{
tempValue =  readByteFromEEPROM(((i*12)+id), 0, chipAddres);
```

```
if(tempValue != keyArray[id]) {isFind=0; break; }
 }
if(isFind)
{
storeSruct->findIndex = i;
storeSruct->findPresence = 1;
storeSruct->statusFindValue = status;
return ;
 }
 }
 }
 }
 }
}
 byte readByteFromEEPROM(int addres, byte bitAddres, byte chipAddres)
{
return EEPROM.read(addres);
}
void updateByteToEEPROM(int addres, byte bitAddres, byte chipAddres, byte value)
{
return EEPROM.update(addres, value);
}
```

Додаток Б – Схема з’єднань КФС



Додаток В – Схема підключення безперебійника з захистом від розряду



Додаток Г – Скетч програми для тестування PIR-датчика

```
#define PIR_AOUT A0  // PIR analog output on A0

#define PIR_DOUT 2   // PIR digital output on D2

#define LED_PIN  13  // LED to illuminate on motion

#define PRINT_TIME 100 // Rate of serial printouts

unsigned long lastPrint = 0; // Keep track of last serial out

void setup()

{  Serial.begin(115200);  // Serial is used to view Analog out

  // Analog and digital pins should both be set as inputs:

  pinMode(PIR_AOUT, INPUT);

  pinMode(PIR_DOUT, INPUT);

  // Configure the motion indicator LED pin as an output

  pinMode(LED_PIN, OUTPUT);

  digitalWrite(LED_PIN, LOW); // Turn the LED off}

void loop()

{  // Read OUT pin, and set onboard LED to mirror output

  readDigitalValue();

  // Read A pin, print that value to serial port:

  printAnalogValue();}

void readDigitalValue()

{  // The OpenPIR's digital output is active high

  int motionStatus = digitalRead(PIR_DOUT);

  // If motion is detected, turn the onboard LED on:

  if (motionStatus == HIGH)

    digitalWrite(LED_PIN, HIGH);

  else // Otherwise turn the LED off:

    digitalWrite(LED_PIN, LOW);}

void printAnalogValue()

{  if ( (lastPrint + PRINT_TIME) < millis() )  {    lastPrint = millis();    // Read in analog value:

    unsigned int analogPIR = analogRead(PIR_AOUT);

    // Convert 10-bit analog value to a voltage

    // (Assume high voltage is 5.0V.)

    float voltage = (float) analogPIR / 1024.0 * 5.0;

    // Print the reading from the digital pin.

    // Mutliply by 5 to maintain scale with AOUT.

    Serial.print(5 * digitalRead(PIR_DOUT));

    Serial.print(',');    // Print a comma

    Serial.print(2.5);    // Print the upper limit

    Serial.print(',');    // Print a comma

    Serial.print(1.7);    // Print the lower limit

    Serial.print(',');    // Print a comma

    Serial.print(voltage); // Print voltage

    Serial.println(); }}
```

Додаток Д - Ревізійна анкета для аудиту безпеки системи

№ з/п	Запитання для аудиту	Відповідь R_i (0/1)	Значення P_i
1	Чи часто проводиться перевірка роботи датчиків та інших комплектуючих?	0	2
2	Чи часто відбувається поновлення комплектуючих елементів?	0	3
3	Чи часто відбувається оновлення програмного забезпечення системи?	0	2
4	Чи має система захист від побічного електро-магнітного випромінювання?	0	3
5	Чи конфіденційний канал зв’язку?	1	4
6	Чи передбачено доступ за допомогою смарт-носіїв?	1	3
7	Чи передбачено можливість віддаленого керування системою?	1	4
8	Чи передбачено можливість відправлення сповіщень про втручання та ін. системою?	1	4
9	Чи забезпечена система резервним джерелом живлення?	1	4
10	Чи має система захист від фізичного втручання до апаратного забезпечення?	0	4
11	Чи є можливість доповнення системи?	1	2

Додаток Е – Розрахунок зважених значень міри для можливих загроз

№	Сценарій загрози	Активи		Вразливості		$CM_{s,j=dp1}$	$CM_{s,j=dp2}$
		КЦД	A(a)	АПН	V(v)		
S_1	Критичний збій в роботі КФС	Д	a_1=3	А	v_6=4	2	1
S_2	Втрата живлення від мережі	Д	a_5=4	А	v_7=4	3	2
S_3	Відсутність покриття мережі GSM	Д	a_3=4	А	v_8=4	1	1
S_4	Поломка давачів, або комплектуючих системи	Д	a_5=4	П	v_{10}=4	3	4
S_5	Поломка давачів, або комплектуючих системи	Ц	a_5=2	П	v_{10}=4	3	3
S_6	Фізичні пошкодження апаратного забезпечення	Д	a_5=4	Н	v_{11}=4	1	4
S_7	Спотворення показів датчиків	Ц	a_2=4	Н	v_3=3	1	3
S_8	Атаки на смарт-карти	К	a_4=2	Н	v_5=3	2	3
S_9	Атаки на смарт-карти	Д	a_4=4	Н	v_5=3	2	1

Додаток Є – Результати оцінки рівня загроз та рівня ризику

Загроза	Вимоги впливу	Рівень вразливості	Рівень загрози	Рівень ризику
Витік інформації	5	5	25	Низький
Отримання контролю над системою	5	10	90	Високий
Помилки введення	9	7	63	Середній
Спотворення інформації	9	8	72	Високий
Зміна, спотворення показів датчиків	9	10	90	Високий
Збій у роботі КФС	10	9	90	Високий
Збої в роботі мережі енергопостачання	10	5	50	Середній
Збій мережі GSM зв’язку	10	10	100	Високий
Видалення інформації	10	8	80	Високий
Фізичний, руйнівний вплив на апаратне забезпечення КФС	10	10	100	Високий
Технічні порушення роботи датчиків та іншого апаратного забезпечення	10	9	90	Високий

Додаток Ж

УДК 004.056

І.В. Ярощук – ст. гр. СБм-61, Ю.Л. Скоренький к.ф.-м.н., доц.
(Тернопільський національний технічний університет імені Івана Пулюя)

РИЗИК-ОРІЄНТОВАНИЙ ПІДХІД ДЛЯ РОЗРОБКИ БЕЗПЕЧНИХ КІБЕРФІЗИЧНИХ СИСТЕМ НА БАЗІ ARDUINO

UDC 004.056

I. Yaroshchuk, Dr. Yu. Skorenkyy
(Ternopil Ivan Puluj National Technical University)

RISK-ORIENTED APPROACH FOR DEVELOPING SECURE ARDUINO-BASED CYBERPHYSICAL SYSTEMS

Ключові слова: інформаційна безпека, кіберфізичні системи, оцінка ризиків, Arduino.
Key words: informational security, cyberphysical systems, risk assessment, Arduino.

В Україні та світі масово розвиваються системи IoT, з ними розробляється величезна кількість кіберфізичних систем. Під терміном кіберфізичні системи слід розуміти сукупність обчислювальних та фізичних складових, які спроєктовані та взаємодіють між собою як єдина система, що може адаптуватись до змін фізичного середовища [1]. Ці системи проникли майже у всі сфери життєдіяльності людини, де надають нові функціональні можливості, що покращують якість життя та технологічні процеси в різних сферах.

Arduino є однією з найпопулярніших фізично-програмних платформ [1, 2] і являє собою невеликий електронний пристрій на друкованій платі, який дає змогу керувати великою множиною датчиків, електродвигунами, освітленням а також забезпечує можливість передачі та отримання інформації. Платформа Arduino це готовий електронний блок [2, 3] для якого доступне спеціалізоване програмне забезпечення. Мікроконтролери Arduino завдяки своїй доступності, практичності та великій множині сфер застосування можна легко використовувати для розробки КФС.

При розробці кожної системи, в тому числі кіберфізичної, потрібно враховувати всі ризики і загрози, які можуть виникнути після введення системи в експлуатацію. Ризики для безпеки спричинені взаємодією між середовищем та КФС, всередині КФС і між КФС та авторизованими користувачами. Оцінка та управління ризиками зосереджені на виявленні вразливих місць в системі та оцінці можливих збитків.

На основі зробленого аналізу можна стверджувати, що основні ризики для КФС спричинені вразливостями платформи, мережі, програмного та апаратного забезпечення, а також технічними вразливостями та вразливостями управління. Формалізацію якісної оцінки доцільно проводити, спираючись на досвід експертів, тоді як кількісна оцінка має бути побудована на аналізі об’єктивних числових даних.

В доповіді виділено та охарактеризовано методології оцінки ризику для забезпечення безпечної розробки кіберфізичних систем.

Література.

1. Alur R. Principles of Cyber-Physical Systems. MIT Press, 2015. - 464 p.
2. Ziemann V. A. Hands-On Course in Sensors Using the Arduino and Raspberry Pi. Boca Raton: CRC Press, 2018. - 258 p.
3. Гаврілов Д. В., Осадчук О. В., Звягін О. С. Основи комп’ютерного проектування та моделювання РЕА. Лабораторний практикум. Частина 1 – Вінниця : ВНТУ, 2015. – 99 с.

Додаток З

УДК 621.326
Ярощук І. – ст. гр. СБм-51
Тернопільський національний технічний університет імені Івана Пулюя

МОЖЛИВОСТІ ЗАСТОСУВАННЯ ПІРОМЕТРИЧНИХ ДАТЧИКІВ НА ПЛАТФОРМІ ARDUINO ДЛЯ КОНТРОЛЮ ПЕРИМЕТРУ

Науковий керівник: к.ф.-м.н., доц. Скоренький Ю.Л.

Yaroshchuk I.
Ternopil Ivan Pul'uj National Technical University

POSSIBILITIES OF APPLICATION OF PYROMETRIC SENSORS ON THE ARDUINO PLATFORM FOR PERIMETER CONTROL

Supervisor: Dr. Yu. Skorenkyy

Ключові слова: інформаційна безпека, пірометричні датчики, Arduino.
Keywords: informational security, pyrometric sensor, Arduino.

В Україні та світі доволі швидкими темпами впроваджують найновіші досягнення телекомунікаційних і комп'ютерних технологій. У фінансових, промислових, торгівельних та соціальних сферах активно впроваджуються телекомунікаційні та кіберфізичні системи, технології інтернету речей, сенсорні мережі. У зв'язку з цим зростає інтерес до проблем інформаційної безпеки. Класифікація загроз безпеці інформації є доволі широкою, тому в даному дослідженні приділена особлива увага загрозам штучного походження, викликаними навмисними діями зловмисників та порушників, а саме контролю фізичного доступу в контрольовану зону, обмежену охоронюваним периметром. Щоб забезпечити захист такої системи від фізичного впливу зловмисника, було запропоновано використання пірометричних датчиків на базі платформи Arduino для інфрачервоного контролю периметру та попередженню несанкціонованого доступу.

Arduino є однією з найпопулярніших фізично-програмних платформ [1, 2] і являє собою невеликий електронний пристрій на друкованій платі, який дає змогу керувати великою множиною датчиків, електродвигунами, освітленням а також забезпечує можливість передачі та отримання інформації. Платформа Arduino це готовий електронний блок [2, 3] для якого доступне спеціалізоване програмне забезпечення. Під електронним блоком слід розуміти друковану плату з вмонтованим мікроконтролером, мінімальним набором елементів для забезпечення його роботи та роз'ємами, що дозволяють підключати зовнішні пристрої та зв'язок з комп'ютером для здійснення програмування мікроконтролера. мікроконтролера. Не менш важливою складовою є програмне забезпечення, що включає в себе доволі просте середовище розробки та С-подібну мову програмування для мікроконтролерів.

Моніторинг було проведено з використанням пірометричних датчиків MLX90614 та HC-SR501. MLX90614 – інфрачервоний термометр для безконтактного вимірювання температури. Такий датчик вимірює дві температури: температуру об'єкта та температуру середовища. Температура об'єкта вимірюється безконтактним способом, а температура середовища вимірюється на кристалі датчика. Температура об'єкта вимірюється в діапазоні від -70 до 380 градусів з 17-бітовим розширенням за допомогою зчитування інфрачервоного випромінювання, що виходить від нього. Така точність дозволяє датчику розрізняти температуру між 25 ° C і 25.02 ° C. У корпусі об'єднані ІЧ-детектор (MLX81101) і мікросхема обробки сигналу (MLX90302). Завдяки застосуванню низькошумового підсилювача, 17-бітного АЦП і потужного DSP процесора датчики мають високу точність і розширення.

Результатом вимірювань є середня температура всіх об'єктів, що потрапляють в робочу область датчика. Точність стандартних моделей становить -0.5 ° C, а точність моделей для медичного застосування (MLX90614ESF-DCI) доходить до 0.2 ° C. Слід враховувати, що дана точність може бути досягнута тільки в тому випадку, якщо датчик знаходиться в стані термічної рівноваги. На його зміни можуть впливати гарячі або холодні об'єкти які знаходяться поруч. Модуль датчика руху HC-SR501 працює на основі піроелектричного ефекту та складається з PIR-датчика 500BP з додатковою електричною розв'язкою на мікросхемі BISS001 та лінзи Френеля, яка використовується для збільшення радіусу огляду та підсилення ІЧ-сигналу. Модуль використовується для виявлення руху об'єктів, що випромінюють тепло (ІЧ-випромінювання). При зміні температури в кристалах датчика виникає електричне поле, і як результат – спрацьовує датчик. Завдяки своїм властивостям та відносно невисокій ціні дані датчики та їх модифікації є дуже зручними для використання.

В ході виконання дослідження було розроблено модель та прототип, які дозволили теоретично обґрунтувати та експериментально підтвердити застосовність пірометричних датчиків бюджетного класу, підключених до плати Arduino, виявляти рух теплової цілі в зоні дії датчика, обмеженої розмірами приміщення, та продукувати інформаційні сигнали або ініціювати дію актуаторів [4], які можуть блокувати загрозу несанкціонованого доступу до захищеної інфраструктури. Побічним результатом розробки є можливість інтеграції протипожежної сигналізації, яка за умови достатньої кількості та оптимального розподілу піроелектричних давачів може бути значно більш чутливою, ніж стандартні рішення. Досліджувалася також можливість застосування системи для вимірювання температури тіла людини, однак слід зазначити, що давачі бюджетного класу мають доволі низьку точність та потребують окремого калібрування, що робить визначення температури для цілей медичної діагностики неможливим, якщо обмежуватися лише пірометричними датчиками MLX90614, HC-SR501 та спорідненими до них. Практичне застосування може мати комбінована система, в якій прецизійний інфрачервоний датчик [5] направляється на об'єкт, виявлений за допомогою пристрою, розробленого в даному дослідженні. В цьому випадку висока точність та мала кутова роздільна здатність прецизійного датчика доповнюються ширококутним та безінерційним датчиком розробленої установки, актуатори (наприклад, мікросервомотор) керуються програмою платформи Arduino.

В підсумку можна стверджувати, що програмно-апаратні засоби типу вимірювального тепловізійного комплексу на платформі Arduino придатні як для моделювання і прототипування, так і для практичного вирішення задачі охорони периметра від несанкціонованого фізичного втручання сторонніх осіб.

1. Alur R. Principles of Cyber-Physical Systems // MIT Press, 2015. - 464 p.
2. Ziemann V. A Hands-On Course in Sensors Using the Arduino and Raspberry Pi // Boca Raton: CRC Press, 2018. - 258 p.
3. Основи комп'ютерного проектування та моделювання РЕА. Лабораторний практикум. Частина 1: лабораторний практикум / Д. В. Гаврілов, О. В. Осадчук, О. С. Звягін – Вінниця : ВНТУ, 2015. – 99 с.
4. Lee E.A., Seshia S.A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach (second edition) // MIT Press, 2017. - 564 p.
5. Hikvision Thermal Products [Електронний ресурс]. Режим доступу: https://www.hikvision.com/en/products/Thermal-Products/