МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

"На правах рукопису"
УДК 681.3.06

«До захисту допущено»
Завідувач кафедри
_____ Шуклін Г.В.
(підпис) (ініціали, прізвище)
“____” _____ 2022р.

МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 Кібербезпека

на тему: **ФОРМУВАННЯ КОДОВИХ СТРУКТУР НА БАЗІ ВИПАДКОВИХ ЗАВАД ТА КРИПТОАНАЛІЗ ЇХ ВЛАСТИВОСТЕЙ**

Студент групи СЗЗМ – 71 Правдивий Олександр Андрійович
(підпис)

Керівник д.т.н., доц. Ахрамович Володимир Миколайович
(підпис)

Нормоконтроль: ст. викладач Гребенніков Асаді Болдгоягович
(підпис)

Київ – 2022

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

_____ Шуклін Г.В.

(підпис) (ініціали, прізвище)

“__” _____ 2022р.

ЗАВДАННЯ

на атестаційну роботу магістра

студенту Правдивому Олександру Андрійовичу

1. Тема роботи: Формування кодових структур на базі випадкових завад та криптоаналіз їх властивостей, керівник Ахрамович Володимир Миколайович, д.т.н., доцент., затверджені наказом вищого навчального закладу від “16” лютого 2022 року № 22.

2. Термін здачі студентом оформленої роботи “30 ” травня 2022 р.

3. Об’єкт дослідження: криптографічні методи захисту інформації.

4. Предмет дослідження: системи дистанційного управління з різними видами доступу та ідентифікації.

5. Мета роботи: Формування кодових структур на базі випадкових завад демонстрація переваги їх криптографічних властивостей у порівнянні з алгоритмічним способом генерації із застосуванням натурних випробувань.

6. Перелік питань, які мають бути розроблені:

1) огляд існуючих криптосистем;

2) огляд способів підтвердження інформації;

3) огляд існуючих систем побудови дистанційного управління;

4) програмно-апаратний комплекс дистанційного підключення;

5) порівняльний аналіз розробленої схеми з вже існуючими аналогами.

7. Перелік ілюстративного матеріалу:

Презентація виконана на 10 слайдах для подання за допомогою оверхедів (світлопроекторів) та комп’ютерних засобів.

8. Дата видачі завдання “2” березня 2022 р.

Керівник: Ахрамович Володимир Миколайович _____

Завдання прийняв до виконання: Правдивий Олександр Андрійович _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	**Примітка**
1	Підбір науково-технічної літератури	10.03.22	Виконано
2	Обґрунтування актуальності теми роботи	16.03.22	Виконано
3	Написання першого розділу роботи	до 04.04.22	Виконано
4	Написання другого розділу роботи	до 25.04.22	Виконано
5	Написання третього розділу роботи	до 06.05.22	Виконано
6	Перевірка роботи на плагіат + передзахист	До 01.06.22	Виконано
9	Захист роботи	13.06.22	
10	Випуск	30.06.22	

Студент О. А. Правдивий

Керівник роботи В. М. Ахрамович

РЕФЕРАТ

Правдивий О.А. „ Формування кодових структур на базі випадкових завад та криптоаналіз їх властивостей ”.

Магістерська атестаційна робота зі спеціальності 125 «Кібербезпека»,

рівень освіти другий «Магістр», галузь знань 12 «Інформаційні технології», – Державний університет телекомунікацій: Київ, 2022.

В магістерській атестаційній роботі розглянуто основні криптографічні методи захисту інформації, зокрема принципи організації та роботи, а також системи дистанційного управління з різними видами доступу та ідентифікації.

Робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел і додатків: 87 с., 16 рис., 2 табл., 15 джерел.

У вступі обґрунтовується актуальність теми та формуються завдання дослідження.

У першому розділі розглядаються криптографічні методи шифрування інформації та області їх використання.

У другому розділі розглядаються наявні системи дистанційного управління та їх функціональні можливості та вразливості.

У третьому розділі ведеться розробка системи дистанційного підключення з фізичним формуванням шифрованих ключів.

У четвертому розділі розглядається порівняльна характеристика систем дистанційного підключення з фізичним та алгоритмічним методами утворення ключів.

Ключові слова: симетрична та асиметрична криптосистема, електронно-цифровий підпис (ЕЦП), система дистанційного управління, оперативний запам’ятовуючи пристрій (ОЗП), ідентифікація, KeeLog, Touch Memory.

ЗМІСТ

ВСТУП

З розвитком сучасних інформаційних технологій у повсякденному житті людини з'являються все новіші та зручніші пристрої, що мають можливість дистанційного керування. До простих пристроїв дистанційного керування можна віднести велику кількість побутових електроприладів таких як, наприклад, телевізор, DVD, аудіосистеми; також до більш складних пристроїв з дистанційним керуванням відносяться охоронні системи приміщень, системи сигналізації автомобілів, системи ППО.

Бездротові радіосистеми безпеки та сигналізації дозволяють організувати багатоканальні системи обмеження доступу, управління та стеження за різноманітними об'єктами. Ці системи широко використовуються для охорони периметрів та окремих об'єктів. Тому закладені у них системи повинні мати гарантовану стійкість по відношенню до несанкціонованого доступу.

Застосування сучасних технологій та методів обробки сигналів робить бездротові системи конкурентоспроможними порівняно з звичайними провідними системами, а іноді, навіть, єдино можливим варіантом організації охорони.

Перед нами стоїть завдання максимально убезпечити свою власність від зловмисника, а також провести порівняльний аналіз схем дистанційного управління. На прикладі автосигналізації розглянемо систему розпізнавання "свій - чужий".

Основою системи є ідея додаткового запиту від наземного передавача, шляхом передачі широкомовного радіосигналу з наземної станції, та отримання відповіді від транспондера (приймача-передавача).

Більшість автосигналізацій зараз управляються з допомогою брелока в радіодіапазоні 433 МГц, тому викрадачам нічого не варто перехопити сигнал брелока та використовувати його для підробки коду зняття сигналізації з охорони.

Навіть якщо сигналізація має алгоритм кодування Keeloq існують спеціальні пристрої та методики, що дозволяють отримати код зняття з охорони. Всякого роду транспордери та мітки також не рятують від сканування чи ретрансляції. Найнадійніший метод (крім пристроїв пізнання відбитка пальця або сітківки ока) - введення коду зняття з охорони, за допомогою набору на клавішній панелі. На жаль, цей спосіб не дуже зручний для відкриття дверей автомашини, але цілком придатний для зняття блокування двигуна та вимкнення режиму "антирозбій".

Я пропоную застосувати додатковий пристрій автосигналізації з генератором шумів для максимального захисту автомобіля.

РОЗДІЛ 1. ОГЛЯД КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ.

Основні поняття.

Криптографічними методами захисту називаються спеціальні методи перетворення інформації, внаслідок яких маскується її зміст.

Виділимо два основні класи криптографічних методів:

1. Симетричні криптосистеми;

2. Асиметричні (криптосистеми з відкритим ключем).

1.1. Симетричні криптосистеми.

У симетричній криптосистемі шифрування використовується один ключ для зашифровування та розшифровування інформації. Це означає, що будь-хто, хто має доступ до ключа шифрування, може розшифрувати повідомлення. Відповідно, з метою запобігання несанкціонованому розкриття зашифрованої інформації всі ключі шифрування в симетричній криптосистемі повинні триматися в секреті. Саме тому симетричні криптосистеми називають криптосистемами з секретний ключем - ключ шифрування повинен бути доступний тільки тим, кому призначено повідомлення. Симетричні криптосистеми називають ще одноключовими криптографічними системами, або криптосистемами із закритим ключем.

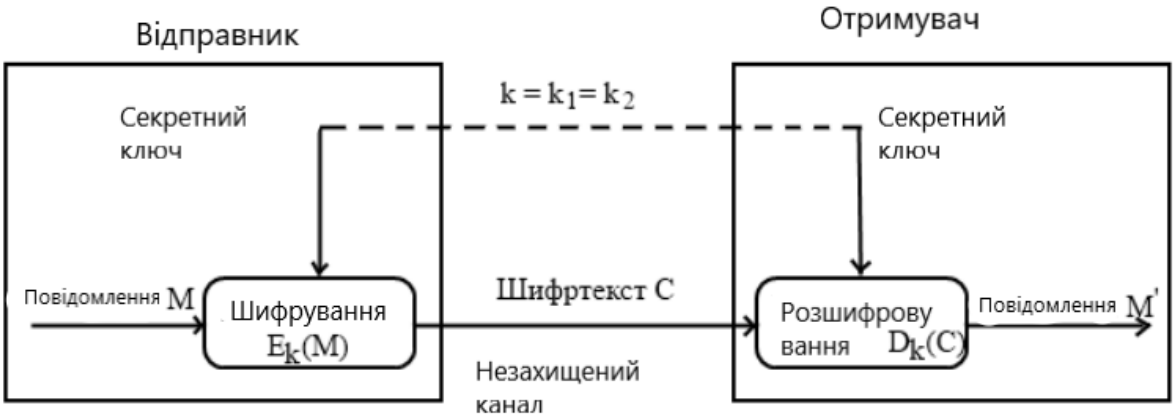

Рисунок 1. Схема симетричної криптосистеми шифрування.

Дані криптосистеми характеризуються найвищою швидкістю шифрування, і з їх допомогою забезпечуються як конфіденційність, так автентичність і цілісність переданої інформації.

Конфіденційність передачі інформації за допомогою симетричної криптосистеми залежить від надійності шифру та забезпечення конфіденційності ключа шифрування. Зазвичай ключ шифрування є файлом або масивом даних і зберігається на персональному ключовому носії, наприклад, дискеті або смарткарті з обов'язковим вживанням заходів, що забезпечують недоступність персонального ключового носія будь-кому, крім його власника.

Автентичність забезпечується за рахунок того, що без попереднього розшифровування практично неможливо здійснити смислову модифікацію та підробку криптографічно закритого повідомлення. Фальшиве повідомлення не може бути правильно зашифровано без знання секретного ключа.

Цілісність даних забезпечується приєднанням до переданих даних спеціального коду (імітовставки), що виробляється за секретним ключа. Імітовставка є різновидом контрольної суми - тобто деякою еталонною характеристикою повідомлення, за якою здійснюється перевірка цілісності

останнього. Алгоритм формування імітовставки повинен забезпечувати її залежність за деяким криптографічним законом від кожного біта повідомлення. Перевірка цілісності повідомлення виконується одержувачем повідомлення шляхом формування імітовставки за секретним ключем, що відповідає отриманому повідомленню, та її порівняння з отриманим значенням імітовставки. При збігу робиться висновок про те, що інформація не була модифікована по дорозі від відправника до одержувача.

Симетричне шифрування ідеально підходить для шифрування інформації «для себе», наприклад, з метою запобігання доступу до неї при відсутності власника. Це може бути як архівне шифрування вибраних файлів, так і прозоре (автоматичне) шифрування цілих логічних чи фізичних дисків.

Маючи високу швидкість шифрування, одноключові криптосистеми. дозволяють вирішувати багато важливих завдань захисту. Проте автономне використання симетричних криптосистем у комп'ютерних мережах породжує проблему розподілу ключів шифрування між користувачами.

Перед початком обміну зашифрованими даними необхідно обмінятись секретними ключами з усіма адресатами. Передача секретного ключа симетричної криптосистеми не може бути здійснена за загальнодоступним каналам зв'язку, тому секретний ключ треба передавати відправнику та одержувачу по захищеному каналу. Для забезпечення ефективного захисту циркулюючих у мережі повідомлень необхідна величезна кількість часто змінних ключів (один ключ на кожну пару користувачів).

Під час передачі ключів користувачам необхідно забезпечити конфіденційність, автентичність та цілісність ключів шифрування, що потребує великих додаткових витрат. Ці витрати пов'язані з необхідністю передачі

секретних ключів по закритих каналах зв'язку або розподілом таких ключів за допомогою спеціальної служби доставки, наприклад, за допомогою кур'єрів.

Проблема розподілу секретних ключів при великій кількості користувачів є дуже трудомістким та складним завданням. У мережі на N користувачів необхідно розподілити N(N-1)/2 секретних ключів, тобто число розподілених секретних ключів зростає за квадратичним законом збільшенням числа абонентів мережі.

Основним недоліком симетричного шифрування є те, що секретний ключ може бути відомий і відправнику, і одержувачу. З одного боку, це ставить нову проблему розсилки ключів. З іншого сторони, одержувач на підставі наявності шифрованого та розшифрованого повідомлення не може довести, що він отримав це повідомлення від конкретного відправника, оскільки таке саме повідомлення він міг згенерувати і сам.

1.2. Асиметричні криптосистеми.

Ще одним великим класом криптографічних систем є так звані асиметричні або дво-ключові системи. Ці системи характеризуються тим, що для шифрування та для розшифрування використовуються різні ключі, пов'язані між собою деякою залежністю. Застосування таких шифрів стало можливим завдяки Д. Шеннону, який запропонував будувати шифр таким способом, щоб його розкриття було еквівалентне рішенню математичного завдання, що вимагає виконання обсягів обчислень, що перевершують можливості сучасних ЕОМ (наприклад, операції з великими простими числами та їх добутками).

Один із ключів (наприклад, ключ шифрування) може бути зроблено загальнодоступним, і в цьому випадку проблема отримання загального секретного ключа для зв'язку відпадає. Якщо зробити загальнодоступним ключ розшифрування, то на основі отриманої системи можна побудувати систему автентифікації переданих повідомлень. Оскільки у більшості випадків один ключ із пари стає загальнодоступним, такі системи отримали також назву криптосистем із відкритим ключем.

Перший ключ не є секретним і може бути опублікований для використання всіма користувачами системи, які зашифровують дані. Розшифрування даних за допомогою відкритого ключа неможливе. Для розшифрування даних одержувач зашифрованої інформації використовує другий ключ, який є таємним. Зрозуміло, ключ розшифрування не може бути визначено з ключа зашифрування.

Криптосистема з відкритим ключем визначається трьома алгоритмами: генерації ключів, шифрування та розшифрування. Алгоритм генерації ключів відкритий, кожен може подати йому на вхід випадковий рядок r належної довжини та отримати пару ключів (k1, k2). Один із ключів (наприклад, k1) публікується, він називається відкритим, а другий, званий секретним,

зберігається у таємниці. Алгоритми шифрування Ek та розшифрування Dk такі, що з будь-якого відкритого тексту m справедливо, що $Dk_1(Ek_2(m))= m$.

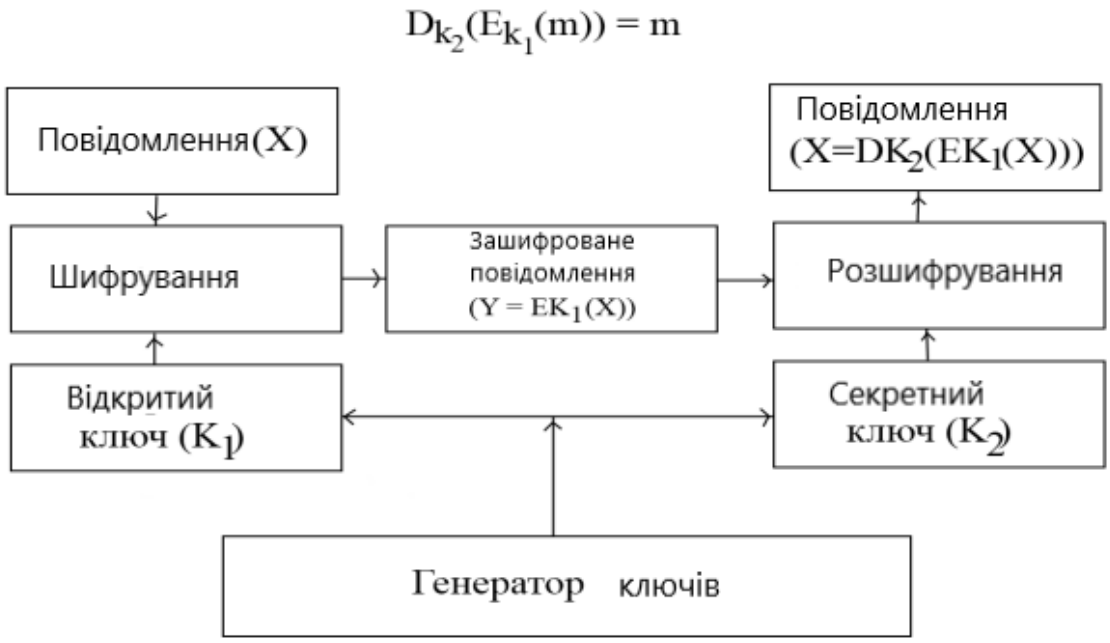

Рисунок 2. Використання асиметричного метода шифрування.

Розглянемо тепер гіпотетичну атаку зловмисника на цю систему. Противнику відомий відкритий ключ k1, але невідомий відповідний секретний ключ k2. Противник перехопив криптограму d і намагається знайти повідомлення m де d = Ek(m). Оскільки алгоритм шифрування відкрито, противник може просто послідовно перебрати всі можливі повідомлення довжини m, обчислити для кожного такого повідомлення mi криптограму di = Ek (mi) і порівняти di і d. Тоді повідомлення, для якого di = d і буде шуканим відкритим текстом. Якщо пощастить, то відкритий текст буде знайдено досить швидко. У найгіршому ж випадку перебір буде виконаний за час порядку 2nT(m), де T(m) – час, потрібне для шифрування повідомлення довжини m. Якщо повідомлення мають довжину близько 1000 бітів, то такий перебір неможливий на практиці яких найпотужніших комп'ютерах.

Такий спосіб атаки на криптосистему та найпростіший алгоритм пошуку відкритого тексту називається алгоритмом повного перебору. Використовується також і інша назва: "метод грубої сили". Інший найпростіший алгоритм пошуку відкритого тексту – вгадування. Цей очевидний алгоритм вимагає невеликих обчислень, але спрацьовує з малою імовірністю (при великих довжинах текстів).

Насправді супротивник може намагатися атакувати криптосистему різними способами та використовувати різні, найбільш витончені алгоритми пошуку відкритого тексту. Зловмисник може спробувати відновити секретний ключ, використовуючи знання (загалом несекретні) про математичну залежності між відкритим та секретним ключами. Природно рахувати криптосистему стійкою, якщо будь-який такий алгоритм вимагає практично нездійсненного обсягу обчислень або спрацьовує з нехтувано малою ймовірністю. Це і є теоретико-складнісний підхід до визначення стійкості. Для його реалізації щодо того чи іншого типу криптографічних систем необхідно виконати таке: дати формальне визначення системи даного типу; дати формальне визначення стійкості системи; довести стійкість конкретної конструкції даного типу.

Тут одразу виникає ряд проблем.

По-перше, для застосування теоретико-складного підходу необхідно побудувати математичну модель криптографічної системи, що залежить від деякого параметра, званого параметром безпеки, який може приймати як завгодно великі значення (зазвичай передбачається, що параметр безпеки може пробігати весь натуральний ряд).

По-друге, визначення стійкості криптографічної системи залежить від того завдання, яке стоїть перед противником, і від того, яка інформація про схему йому доступна. Тому стійкість систем доводиться визначати та досліджувати окремо для кожного припущення про супротивника.

По-третє, необхідно уточнити, який обсяг обчислень можна вважати "практично нездійсненним". Зі сказаного випливає, що ця величина не може бути просто константою, вона має бути представлена функцією від зростаючого параметра безпеки. Відповідно до тези Едмондса алгоритм вважається ефективним, якщо час його виконання обмежено деяким поліномом від довжини вхідного слова (від параметра безпеки). Інакше кажуть, що обчислення з цього алгоритму практично неможливі. При цьому самі криптографічні системи мають бути ефективними, тобто всі обчислення, запропоновані тією чи іншою схемою повинні виконуватися за поліноміальний час.

По-четверте, необхідно визначити, яку ймовірність можна вважати нехтувано малою. У криптографії прийнято вважати таку будь-яку ймовірність, яка для будь-якого полінома p і для всіх достатньо великих m менше або рівна 1/p(m), де m – параметр безпеки. Отже, за наявності всіх зазначених вище визначень, проблема обґрунтування стійкості криптографічної системи зводиться до доказу відсутності поліноміального алгоритму, який вирішує завдання, що стоїть перед супротивником. Але тут виникає ще одна дуже серйозна перешкода: сучасний стан теорії складності обчислень не дозволяє доводити надполіноміальні нижні оцінки складності для конкретних завдань класу, що розглядається. З цього слідує, що на даний момент стійкість криптографічних систем може бути встановлена лише із залученням будь-яких недоведених припущень. Тому основний напрямок досліджень полягає у пошуку найбільш слабких достатніх умов для існування стійких систем кожного із типів. Здебільшого, розглядаються припущення двох типів - загальні (або теоретико-складносні) і теоретико-числові - тобто припущення про складність конкретних теоретико-числових задач.

1.3. Електронно-цифровий підпис.

ЕЦП – це електронно-цифровий підпис. Він замінює власноручний підпис уповноваженої особи та печатку, так як має таку ж юридичну силу, як і дані 2 реквізити паперового документа. Якщо провести аналогію, то електронно-цифровий підпис, так само як і власноручний підпис та друк, призначений для захисту документа(електронного) від підробки. З його допомогою можна перевірити, чи був документ модифікований після підписання і чи є підписант уповноваженою особою – власником сертифіката ключа ЕЦП.

Електронно-цифровий підпис формується шляхом криптографічного перетворення інформації за допомогою закритого ключа електронно-цифрового підпису та спеціального програмного забезпечення. Він дозволяє вдосконалити електронний документообіг та гарантує достовірність документів. За будь-якої зміни вихідного документа ЕЦП стає недійсним.

1.4. Принцип роботи ЕЦП.

Принцип роботи електронного підпису є досить простим. Кожному користувачу, який бажає брати участь в електронному документообігу генеруються 2 ключі – відкритий та закритий. Підписувач формує документ. Потім, на основі закритого ключа, вмісту документа та спеціального програмного забезпечення генерується послідовність символів, яка і є електронною підписом, та підписаний документ надсилається одержувачу. Одержувач документа за допомогою відкритого ключа підписувача виконує зворотне криптографічне перетворення, тим самим перевіряє ЕЦП відправника та засвідчується в тому, що текст документа не був спотворений.

1.5. Область застосування електронно-цифрового підпису.

У зв'язку з тим, що електронно-цифровий підпис дозволяє захистити електронний документ від змін, перевірити його цілісність, робить неможливим відмову від авторства, дає можливість це авторство довести, сфера застосування ЕЦП досить широка. Серед цілей, для яких застосовуються електронні цифрові підписи, можна виділити такі:

- Забезпечення для електронних документів юридичної значущості допомогою електронного підпису;

- Передача обов'язкової бухгалтерської та податкової звітності у електронному вигляді до податкових інспекцій;

- Обмін електронними документами між організаціями та їх структурними підрозділами;

- Декларування товарів та послуг з метою митного оформлення;

- Авторизація для отримання доступу до спеціалізованих інформаційних ресурсів;

- Передача даних до органів статистики та відділення Пенсійного фонду;

- Подання звітності до контролюючих органів;

- Участь у електронних торгах;

- Захист повідомлень електронної пошти та інше.

РОЗДІЛ 2. ПОБУДОВА СИСТЕМ ДИСТАНЦІЙНОГО УПРАВЛІННЯ.

2.1. Система кодового доступу та ідентифікації KeeLog.

Широку популярність у системах кодового доступу та ідентифікації отримала технологія KeeLoq, яка використовує "стрибаючий" (динамічний) псевдовипадковий код.

Суть методу полягає в тому, що, коли натискається кнопка на пульті дистанційного керування, кодер формує нову комбінацію для передачі, передбачити яку може лише «свій» приймач. При цьому повторна передача коду (наприклад, за допомогою пристрою перехоплення коду) не викликає спрацювання системи. Ключі KeeLoq мають 4,6 мільярда кодових комбінацій, для перебору яких потрібно більше 15 років.

Типові застосування технології KeeLoq:

- автомобільні охоронні системи та іммобілайзери;

- системи ідентифікації;

- системи обмеження доступу та електронні замки;

- пристрої дистанційного керування.

Кодер (передавач).

Формувачі "стрибаючого" коду KeeLoq призначені для односпрямованих систем дистанційного керування. Ініціалізація кодера на передачу коду, відбувається після натискання на кнопку пульта дистанційного керування. У функції кодера входить лише формування кодової посилки, розробнику системи дистанційного управління необхідно подбати про організацію каналу зв'язку. В як канал зв'язку можна використовувати провідний зв'язок, ІЧ-канал або радіоканал.

Розглянемо побудову системи, який використовує радіоканал. На рисунку 3 наведено схему кодера з використанням IMC HCS410 (транспондер фірми Microchip) та передавача RT5-433 на частоту 433.92МГц (Виробництво фірми Telecontrolli).

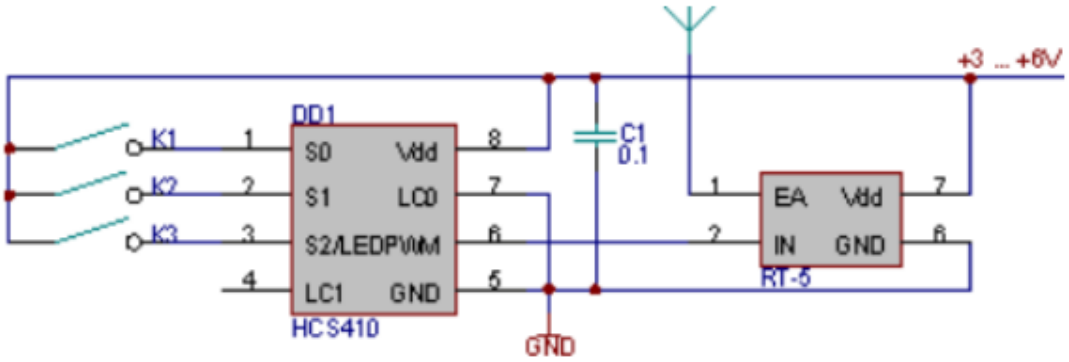

Рисунок 3. Схема кодера.

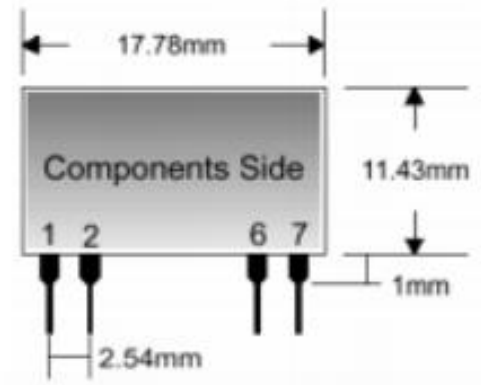

Рисунок 4. Розміри передавача.

У технології KeeLoq використовується своєрідна система реверсивної ідентифікації за принципом «свій – чужий». Розробник системи повинен задати серійний унікальний номер передавача та заводський ключ приймача на основі яких формується секретний ключ, який записується в кодер (передавач) на етапі програмування.

Секретний ключ не може бути зчитаний з кодера, і він ніколи не передається каналом зв'язку. При кожній ініціалізації кодера (натискання кнопки пульта дистанційного керування) формується кодова послідовність, в яку входить 32-бітний «стрибаючий» код отриманий із 64-бітного секретного ключа. Код, що передається, унікальний для кожної нової кодової послідовності. У переданій кодовій посилці передається інформація про натиснуті кнопки та стан батареї харчування.

З метою енергозбереження, у схему кодера включено комутатор живлення формувача кодової послідовності. Якщо жодна із кнопок не натиснута, то формувач знеструмлений. У разі тривалого утримання клавіші в натиснутому стані відбувається автоматичне відключення формувача кодової послідовності. Для повторної ініціалізації передачі коду необхідно відпустити та знову натиснути кнопку на пульті.

Усі кодери KeeLoq повторюють передачу кодової послідовності доти, доки залишається натиснутою кнопка або не спрацював захист від розряду батареї. Кодова послідовність завжди передається повністю, навіть якщо кнопка буде відпущена під час передачі. Кодер автоматично передаватиме всю сформовану кодову послідовність та перейде в режим очікування. У кодері передбачена функція захисту від замикання контактів кнопок або короткочасних, хибних натискань на клавішу.

Декодер (приймач).

Декодер KeeLoq призначений для дешифровки команд, що надходять від кодера каналом зв'язку. Після перевірки прийнятого в кодовій послідовності серійного номера і «стрибаючого» коду, декодер на підставі функціонального коду активізує виходи, що відповідають входам кнопок у кодері. Виходи утримуватимуться в активному стані доти, доки натиснута кнопка на кодері.

Для виконання команд декодером йому необхідно задати свій серійний номер та секретний ключ кодера. Розробник системи має задати декодеру унікальний серійний 28/32 бітний номер на етапі його програмування. Секретний 64-бітний ключ формується декодером етапі «навчання» та синхронізації з кодером. Процедура «навчання» кодера та декодера може бути здійснена користувачем системи для необхідної кількості передавачів.

Програмну реалізацію алгоритму KeeLoq можна здійснити на основі будь-якого мікроконтролера фірми Microchip. Для зберігання ключів може бути використана внутрішня або зовнішня ЕЕПРОМ. Схема декодера з мікроконтролером PIC16F628 фірми Microchip та приймачем RR8-433 фірми Telecontrolli наведено на рис.3.

PIC16F628 приймає кодову посилку від приймача, порівнює її з очікуваною і при збігу виставляє сигнали, що відповідають натиснутій кнопці. При допомогою кнопки К1 здійснюється «навчання» приймача, тобто. Реєстрація нового ключа (брелока) та їх початкова синхронізація. Кількість ключів, які можуть бути зареєстровані в системі, залежить від розміру внутрішньої/зовнішньої ЕЕПРОМ.

Систему декодування можна побудувати і на спеціалізованих мікросхемах, наприклад HCS500, HCS512, HCS515. Однак застосування в якості декодера мікроконтролера дозволяє розробнику покласти нього ряд додаткових функцій.

Наприклад, зв'язування з комп'ютером через інтерфейс RS-232 для здійснення протоколювання та фіксації часу доступу на об'єкт, що охороняється; вбудовування алгоритму декодування у готовий проект для розмежування доступу до різним функціям приладу, що підвищить стійкість до злому, тощо.

Декодер KeeLoq дозволяє змінювати реєстраційну інформацію про передавачі. Для видалення інформації про всі передавачі, необхідно утримувати активний рівень сигналу на навчальному вході декодера протягом 10 секунд. Допускається циклічний запис даних про новий передавач. При цьому передавач, який був зареєстрований раніше за всіх, буде видалений.

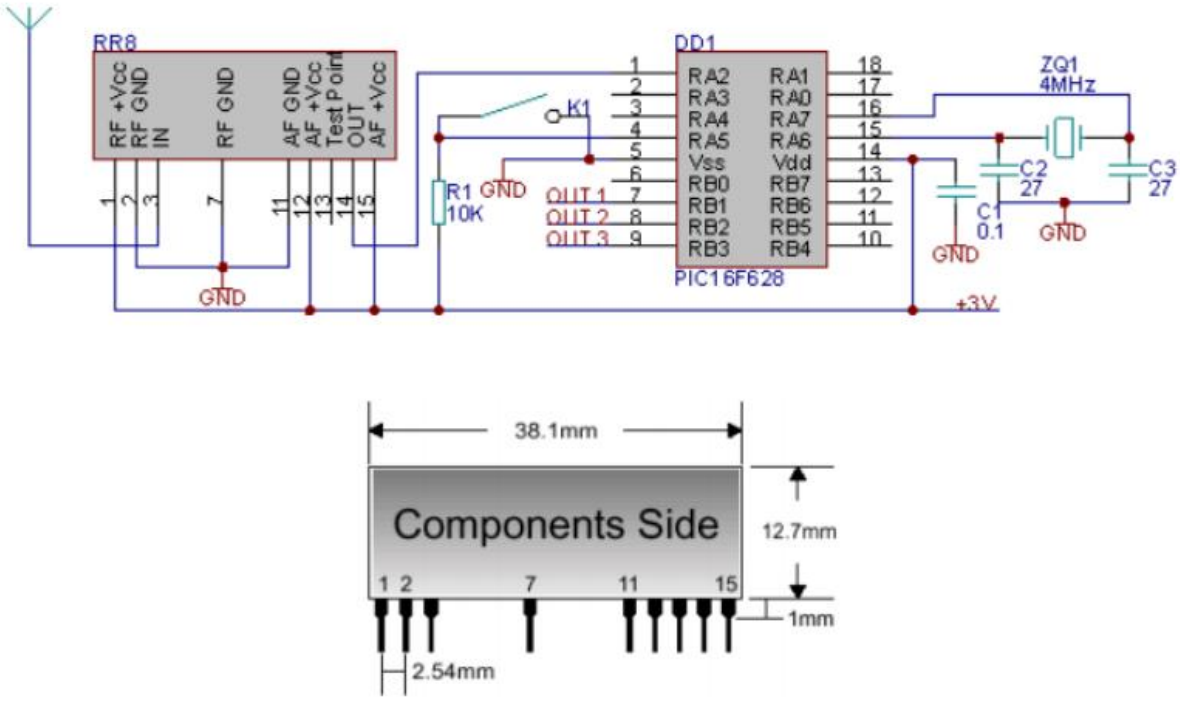

Рисунок 5. Декодер KeeLoq и розміри приймача.

Дальність роботи такої системи становить 30м під час роботи передавача від джерела живлення з напругою 3В. При збільшенні напруги живлення передавача до 12В дальність зв'язку стає вищою 100м.

Для швидкого освоєння технології KeeLoq, фірма Microchip допомогу розробнику випускає ряд комплектів (KeeLoq Evaluation Kit), які включають демонстраційні плати декількох кодерів і декодера, а також все потрібне програмне забезпечення. Згодом ці комплекти можуть використовуватися як програматори ключів для своєї системи.

2.2. Touch Memory – електронний ідентифікатор.

У системах автоматичної ідентифікації персоналу Touch Memory, технічних виробів, товарів найбільш популярними є такі традиційні ідентифікатори, як штрих-код та магнітна смужка.

Однак, незважаючи на простоту та дешевизну, ці ідентифікатори мають ряд суттєвих обмежень. До їхніх недоліків можна віднести незначну інформаційну ємність, неможливість оперативної зміни записаних даних, більшу залежність від умов експлуатації, а також необхідність використання спеціальних пристроїв зчитування, що перетворюють оптичні або магнітні сигнали у цифровий код.

Широке впровадження інформаційних систем у виробництві, управлінській діяльності, фінансової галузі, торгівлі, соціальної сфері потребує створення більш досконалих засобів автоматичної ідентифікації.

До таких засобів можна з повною підставою віднести принципово новий тип електронних ідентифікаторів американської компанії "Dallas Semiconductor". Прилади сімейства DS199X, що отримали назву Touch Memory, мають цілу низку унікальних особливостей.

Touch Memory є енергонезалежною пам'яттю, розміщеною в металевому корпусі, з одним сигнальним контактом та одним контактом заземлення. Корпус нагадує мініатюрну ґудзикову батарейку, що легко кріпиться на виробі або на носії (картка, брелок). Інформація записується та зчитується з пам'яті приладу простим дотиком зчитувального пристрою корпусу Touch Memory.

До сімейства Touch Memory входять 5 (п’ять) приладів, ідентичних по конструкції корпусу, але різних за функціональними можливостями, об'ємом пам'яті, і навіть шляхом доступу до неї (таблиця 1).

Табл. 1. Сімейство Touch Memory

Тип приладу	Унікальний серійний номер	Об’єм блокнотної пам’яті в байтах	Годинник\Таймер	Об’єм основної пам’яті в байтах	Захист доступу до пам`яті	Конструкція корпусу
DS1990A	+	-	-	-	-	F5/F3
DS1991	+	64	-	192	+	F5
DS1992	+	32	-	128	-	F5
DS1993	+	32	-	512	-	F5
DS1994	+	32	+	512	-	F5

У структурі Touch Memory можна виділити чотири основні блоки: постійний запам'ятовуючий пристрій, блокнотну пам'ять, оперативний запам'ятовуючий пристрій, годинник реального часу (для DS1994), а також елемент живлення – вбудовану мініатюрну літієву батарейку. Кожен прилад Touch Memory містить постійний запам'ятовуючий пристрій (ПЗП), в якому зберігається 64-розрядний код, що складається з 8-розрядного коду типу приладу, 48-розрядного унікального серійного номера та 8- розрядної контрольної суми. Дані, що розміщуються в ПЗП, є унікальною кодовою комбінацією, яка записується в прилад за допомогою лазерної установки під час його виготовлення та не може бути змінена протягом усього терміну служби приладу. У процесі запису та тестування на заводі гарантується, що не буде виготовлено два прилади з однаковими номерами.

Оскільки при читанні даних із ПЗП у будь-який момент можливе порушення електричного контакту пристрою зчитування з корпусом приладу, то необхідно контролювати цілісність даних, що зчитуються. Для цієї мети в Touch Memory використовується контроль циклічно надлишковим кодом (CRC).

Попередньо вирахувана контрольна сума молодших 7 байтів вміст ПЗП зберігається у старшому байті. При читанні даних із ПЗП в зчитувальному пристрої (персональна ЕОМ, мікропроцесорний контролер) обчислюється контрольна сума, яка порівнюється з контрольним кодом, записаним у старшому байті. У тому випадку, якщо коди збіглися, серійний номер вирахуваний правильно. Інакше виконується повторне читання даних із ПЗП.

Напруга живлення ПЗП подається по сигнальній лінії даних, що дозволяє, по-перше, заощадити, енергію вбудованої літієвої батареї, і, по-друге, зчитувати пам'ять завжди незалежно від енергії батарейки.

2.3. Оперативний запам'ятовуючий пристрій.

Найпростіший прилад сімейства DS1990 містить лише постійну пам'ять. Всі інші прилади мають у своєму складі також статичну оперативну пам'ять. Число циклів запису-читання в цю пам'ять не обмежене. Живлення пам'яті забезпечується мініатюрною літієвою батарейкою, термін служби якої – 10 років.

Вся оперативна пам'ять поділена на окремі сторінки обсягом по 32 байт. DS1992 містить 4 сторінки, які забезпечують зберігання 256 байтів, DS1993 та DS1994 – 16 сторінок, що дозволяє зберігати 512 байтів. DS1994 містить додаткову 17 сторінку, яка має обсяг 30 байтів і призначена для роботи годинника реального часу (рисунок 6).

Рисунок 6. Оперативний запам’ятовуючий пристрій.

Оскільки дані записуються в пам'ять у момент дотику зчитувача пристрою та корпусу приладу, то порушення електричного контакту в цей момент може призвести до руйнування інформації в пам'яті. Щоб запобігти руйнуванню інформації, у структурі Touch Memory передбачено додаткову буферну пам'ять, яка виконує функцію блокнотної області. Ця пам'ять захищає прилад від випадкового запису нових даних на місце наявних або від запису не за тією адресою.

Об'єм блокнотної пам'яті дорівнює обсягу сторінки оперативної пам'яті – 32 байт для DS1992-94.

Розглянемо принцип роботи блокнотної пам'яті. Всі вхідні дані спочатку записуються в блокнотну пам'ять. Потім вони передаються з неї в пристрій для читання, де порівнюються з даними, які потрібно було записати. Після верифікації виконується операція копіювання вмісту блокнотної пам'яті основну. Оскільки копіювання виконується всередині Touch Memory, то гарантується цілісність інформації навіть у разі порушення зовнішнього контакту.

2.4. Оперативний пристрій із захистом доступу.

Прилади DS 1992-94 мають ідентичну структуру оперативної пам'яті, будь-яка сторінка якої доступна як за читанням (безпосередньо), так і за записом (через блокнотну пам'ять).

Прилад DS 199.1 має більш складну оперативну архітектуру пам'яті. У ньому реалізовано на апаратному рівні захист пам'яті від несанкціонований доступу. Вся енергонезалежна пам'ять поділена на чотири незалежні сторінки по 64 байти, одна зі сторінок – блокнотна пам'ять. Кожна сторінка основної пам'яті складається з 48 байтів, призначених для зберігання даних, та двох службових полів по 8 байтів для зберігання ідентифікатора та пароля (рисунок 7).

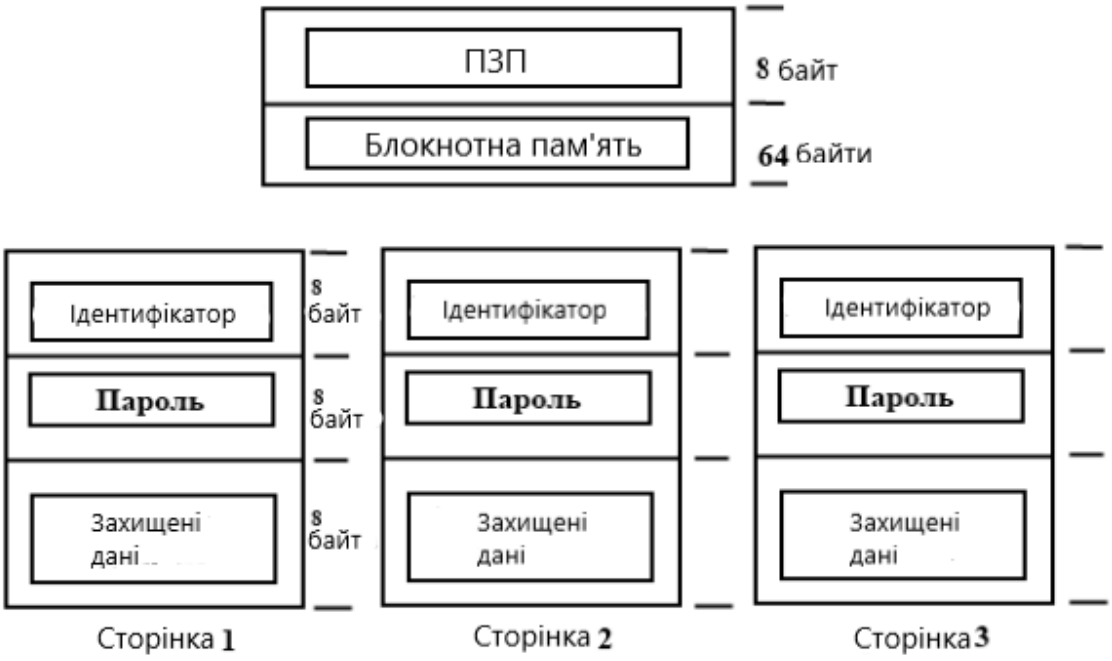

Рисунок 7. Схема зберігання ідентифікатора і пароля.

Механізм доступу до пам'яті реалізований за допомогою двох ключів: відкритого, що зберігається в полі ідентифікатора, та закритого, записаного в полі пароля. Відкритий ключ записується та зчитується, закритий тільки встановлюється і може бути прочитаний. Закритий ключ забезпечує санкціонований доступ до пам'яті та захищений від випадкової зміни за допомогою відкритого ключа.

При початковому форматуванні у службові поля кожної сторінки записуються коди відкритого та закритого ключів даної сторінки. При будь-якому зверненні до пам'яті DS1991 спочатку передається закритий ключ цієї сторінки. Якщо він збігається з ключем, попередньо записаним у полі пароля, пам'ять буде доступна як на запис, так і на читання. При розбіжності кодів дані в пам’ять не записуються, а в режимі читання з DS1991 зчитується послідовність випадкових чисел.

Для запису нового значення закритого ключа в DS1991 необхідно надіслати код відкритого ключа вибраної сторінки. В випадку збігу цього коду з кодом, раніше записаним у полі ідентифікатора, у службове поле даної сторінки записуються нові значення обох ключів, а область даних стирається. При розбіжності кодів значення закритого ключа не змінюється. Реалізований у DS1991 механізм доступу до пам'яті забезпечує надійний захист пам'яті від несанкціонованого запису, що в ряді застосувань вкрай важливо.

2.5. Однопровідний інтерфейс.

Відмінною особливістю Touch Memory є розроблений фірмою "Dallas Semiconductor" протокол обміну зі зчитуючим пристроєм.

Для прийому-передачі інформації використовується одна двонаправлена сигнальна лінія (другий провід –контакт заземлення). Обмін по одній лінії здійснюється в режимі напівдуплексу (або прийом або передача). Взаємодія приладів за однопровідним інтерфейсом організована за принципом “master-slave”. При цьому зчитуючий пристрій завжди «master», а один або кілька приладів Touch Memory – «slave». Взаємодія кількох приладів зі зчитуючим пристроєм по одній двонаправленій лінії підтримується апаратними засобами Touch Memory.

Протокол обміну за однопровідним інтерфейсом є дворівневим. На першому - логічному рівні для взаємодії пристроїв використовуються команди обміну з ПЗП та ОЗП (таблиця 2).

Табл.2. Обмін ПЗП и ОЗП.

Тип пристрою	Команди ПЗП	Команди блокнотної пам’яті	Команди ОП		Команди встановлення паролю
	Читання Пропуск Порівняння Пошук	Читання Запис Копіювання	Читання	Запис	Запис
DS1990A	+	-	-	-	-
DS1991	+	+	+	+	+
DS1992	+	+	+	-	-
DS1993	+	+	+	-	-
DS1994	+	+	+	-	-

Групу команд обміну з ПЗП становлять чотири команди: читання ПЗП, пропуск, порівняння та пошук. Дві останні команди забезпечують взаємодію по одній лінії декількох Touch Memory зі зчитуючим пристроєм. Команда порівняння ініціює обмін із приладом, серійний номер якого вказано. Команда пошуку дозволяє визначити серійний номер одного із приладів, підключених до двонаправленої лінії. Команди обміну з блокнотною та основною пам'яттю обробляються Touch Memory тільки після виконання однієї з команд обміну з ПЗП. Таким чином, при взаємодії кількох приладів, підключених до однієї лінії, пристрій що зчитує посилає по лінії команду порівняння, за допомогою якої вибирається лише один прилад, що приймає надалі команди обміну з пам'яттю.

Усі команди обміну мають фіксований розмір - один байт, дані представлені 8-розрядними цілими числами. «Master»-пристрій завжди ініціює обмін, надсилаючи команди «slave»-пристрою.

Протокол фізичного рівня використовується передачі команд і даних за однопровідним інтерфейсом. Команди та дані передаються в послідовний код. Для забезпечення цілісності переданої інформації протокол обміну фізично регламентує тимчасові параметри сигналів лінії. Протокол обміну даними складається з трьох основних циклів: ініціалізації, запису та читання.

Цикл ініціалізації є початковим циклом будь-якого інформаційного обміну із Touch Memory. У цьому циклі провідний пристрій опитує лінію, визначаючи присутність у ній Touch Memory. Синхронізація циклу ініціалізації здійснюється негативним імпульсом скидання, що формується «master»-пристроєм. Після посилки сигналу «master»-пристрій звільняє лінію та переходить у режим прийому. У тому випадку, якщо до лінії підключено прилад Touch Memory, він виявляє синхросигнал «slave»-пристрою і після тимчасової паузи посилає йому сигнал упізнання (рис. 8). Цей сигнал у відповідь інформує «slave»-пристрій про те, що є електричний контакт з Touch Memory та можна розпочинати обмін.

Рисунок 8. Передача даних по однопровідній двонаправленій лінії.

Дані передаються по однопровідній двонаправленій лінії протягом дискретних часових інтервалів, які називаються тимчасовими сегментами (типова тривалість – близько 60 мкс). При передачі даних використовується широтно-імпульсний метод кодування, що нагадує азбуку Морзе: протягом одного часового сегмента довгі чи короткі стани логічного нуля на лінії визначають значення переданого розряду. Забезпечується швидкість передачі даних до 16,6 кбіт/сек.

Синхронізація часового сегмента під час запису здійснюється негативним фронтом сигналу, що формує «master»-пристрій. Для передачі в Touch Memory логічної одиниці «master»-пристрій після посилки синхросигналу звільняє лінію. Для запису логічного нуля «master»-пристрій підтримує низький стан лінії протягом всього тимчасового сегмента. Описаний цикл запису повторюється для кожного переданого розряду команди.

На початку циклу читання «master»-пристрій також передає до лінії синхронізуючий сигнал низького рівня, після чого звільняє лінію та переходить у режим прийому. Далі протягом усього часового сегменту стан однопровідної лінії визначається «slave»-пристроєм – Touch Memory. При цьому логічна

одиниця передається високим рівнем, а логічний нуль - низьким рівнем однопровідної лінії протягом усього тимчасового сегмента. Найкращий момент для сприйняття даних «master»-пристроєм – це 8 мкс після початку тимчасового сегмента. Цикл читання одного розряду повторюється доти, доки всі дані не будуть зчитані.

Наприкінці кожного часового сегмента провідний пристрій забезпечує паузу в обміні (момент відновлення), утримуючи лінію у високому стані. Призупинення сеансу зв'язку на будь-який час між тимчасовими сегментами можливе, при цьому на лінії підтримується високий стан. У всіх сеансах зв'язку першим передається молодший значущий розряд даних.

2.6. Конструктивні особливості Touch Memory.

Цілий ряд унікальних властивостей Touch Memory забезпечується завдяки незвичайному корпусу приладу. Кристал пам'яті та мініатюрна літієва батарея вмонтовані в герметичному корпусі з нержавіючої сталі діаметром 16 мм та товщиною 5,8 мм (корпус F5) або 3,2 мм (корпус F3).

Сталевий корпус використовується для здійснення електричних контактів. Корпус приладу аналогічний по конструкції корпусу ґудзикової батарейки. Він складається з обідка з денцем та електрично ізольованої кришки. На відміну від звичайних мікросхем, доступ до вмісту пам'яті приладу здійснюється тільки через дві лінії: земляну та двонаправлену сигнальну. Обідок і денце являють собою земляний контакт, а кришечка виконує функцію сигнального контакту. Корпус може витримати понад 1 млн. механічних підключень без помітного зносу та стирання.

Для зчитування даних із приладів Touch Memory використовується контактуючий пристрій Touch Probe (зонд), який являє собою механічний вузол, що складається з двох штампованих металевих деталей, розділених діелектриком. Форма наконечника зонда зроблена такою, щоб він точно сполучався з круглим корпусом приладу. При цьому поглиблена центральна область виконує функцію сигнального контакту, а його обід служить земляним контактом.

Малі розміри Touch Probe дозволяють вбудовувати його безпосередньо в портативний мікропроцесорний контролер, прикріплювати до будь-якої поверхні або використовувати у вигляді окремого ручного пристрою. Взаємодія з приладом забезпечується миттєвим торканням зонда та корпусу Touch Memory таким чином, що денце приладу контактує з поглибленою центральною областю зонда, а обідок - з бічної поверхнею зонда.

Використання простого за конструкцією електричного інтерфейсу забезпечує високу механічну міцність Touch Memory, оскільки у нього відсутні штирі або контакти, які можна пошкодити.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСА ДП З ФІЗИЧНИМ ФОРМУВАННЯМ ШИФРОВАНИХ КЛЮЧІВ.

У цій роботі ми розробили додатковий метод захисту автосигналізації від злому і перехоплення сигналу, що подається господарем автомобіля.

Суть методу у тому, що щоразу натискається кнопка на пульті дистанційного керування, розроблений нами передавач створює шуми, котрі сприяють захисту посланого сигналу від брелока до автомобіля, тим самим захищаючи від несанкціонованого доступу зловмисників.

Розроблена нами схема складається із трьох основних частин: генератор випадкових чисел, приймальний пристрій, IR передавач сигналізації.

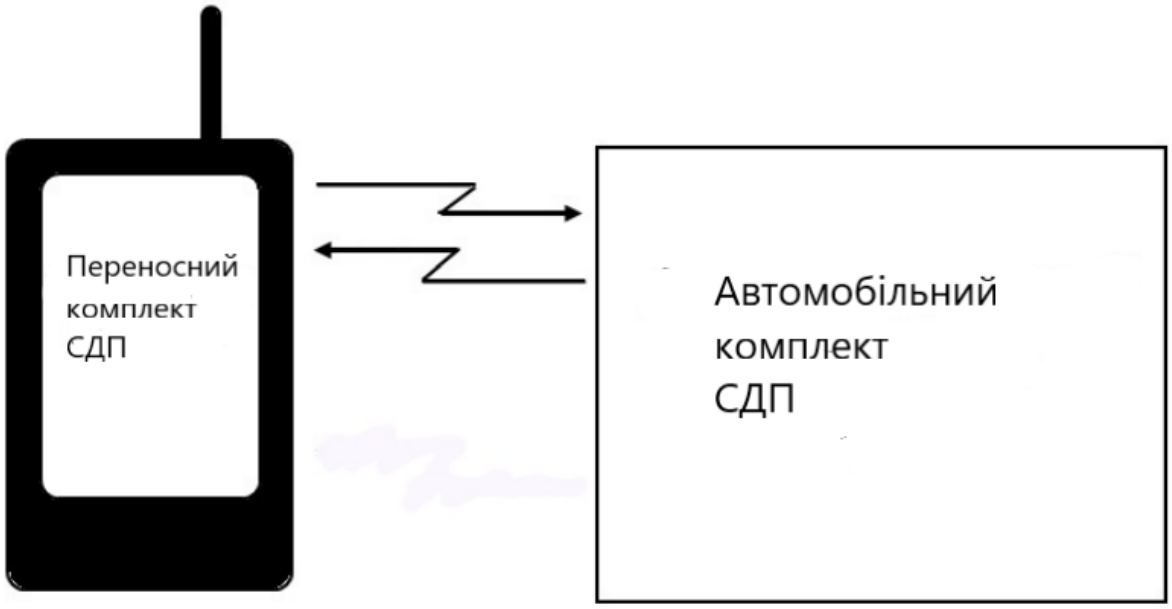

Рисунок 9. Стандартна система сигналізації с дистанційним управлінням по радіоканалу.

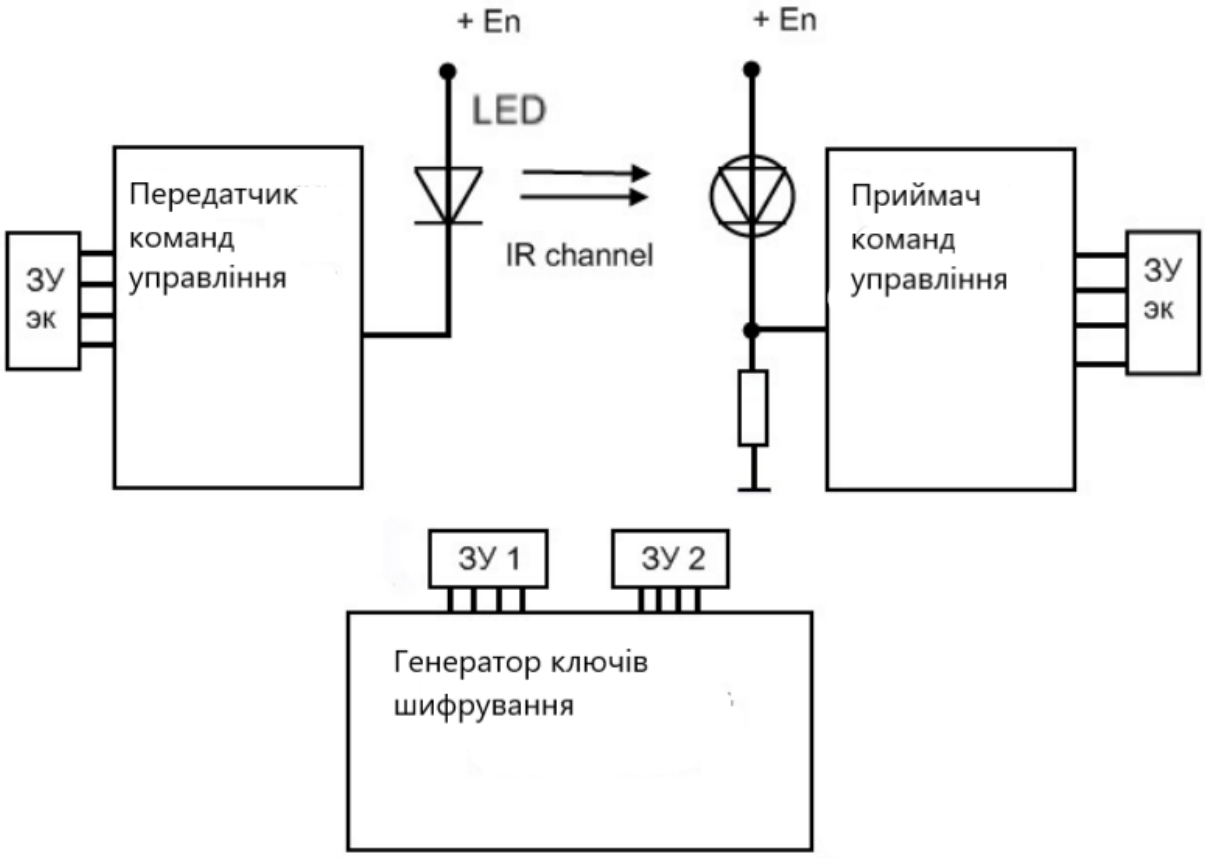

Рисунок 10. Додаткова система с дистанційним управлінням по инфрачервому каналу зв’язку.

Користувач дистанційного керування формує керуючий вплив, що надходить на формувач команд управління.

Команда управління, що є електричним сигналом певної структури, надходить на пристрій узгодження електричних сигналів з каналом зв'язку (модулятор), і далі в канал зв'язку (радіоканал, дротовий, гідроакустичний, ультразвуковий, оптичний канал). Далі сигнал надходить послідовно на приймальний пристрій, дешифратор команд управління, та на формувач керуючих впливів.

Формувач керуючих впливів впливає на об'єкт управління. Зміни в об'єкті управління передаються у формувач сигналів зворотного зв'язку. Сигнали зворотного зв'язку перетворюються та надходять у канал зв'язку, і далі на приймальний пристрій. З приймального пристрою сигнал зворотного зв'язку надходить до оператора дистанційного управління. Сигналом зворотного зв'язку може бути візуальний, акустичний або інший сигнал. Часто в системах дистанційного керування відсутній канал зворотного зв'язку, але у відповідальних системах він обов'язковий.

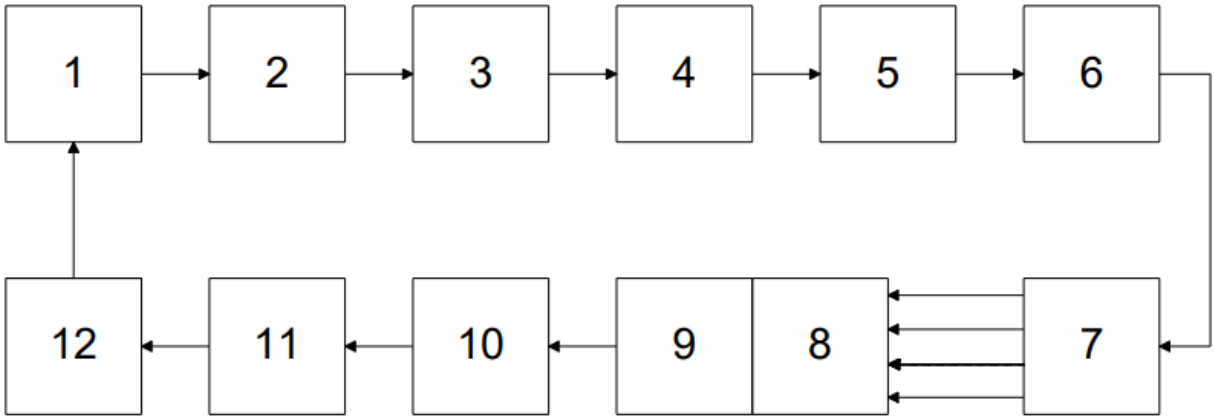

Рисунок 11. Узагальнена структурна схема дистанційного керування.

Цифрами на схемі позначені:

1-оператор дистанційного керування;

2-формувач команд управління;

3-пристрій узгодження електричних сигналів з каналом зв'язку;

4-канал зв'язку;

5-приймальний пристрій;

6-дешифратор команд керування;

7-формувач керуючих впливів;

8-об'єкт управління;

9-формувач сигналів зворотного зв'язку;

10-перетворювач;

11-канал зворотного зв'язку;

12-приймальний пристрій.

Розроблена структурна схема системи дистанційного керування автосигналізацією по радіоканалу представлена на рисунку 12.

Схема працює аналогічно описаній вище з невеликими поправками. Оператор дистанційного керування формує керуючий вплив, що надходить на формувач команд управління. Команда управління, що є електричним сигналом певної структури, надходить на підсилювач потужності, далі сигнал подається на передавач. По радіо каналу сигнал надходить в приймач. Формувач керуючих впливів впливає на об'єкт управління. Зміни в об'єкті управління передаються у формувач сигналів зворотного зв'язку. Сигнали зворотного зв'язку перетворюються та надходять у канал зв'язку, і далі на приймальний пристрій. З приймального пристрою сигнал зворотного зв'язку надходить до оператора дистанційного управління. Сигналом зворотного зв'язку може бути візуальний, акустичний або інший сигнал. Часто в системах дистанційного керування відсутня канал зворотний зв'язок, але у відповідальних системах він обов'язковий.

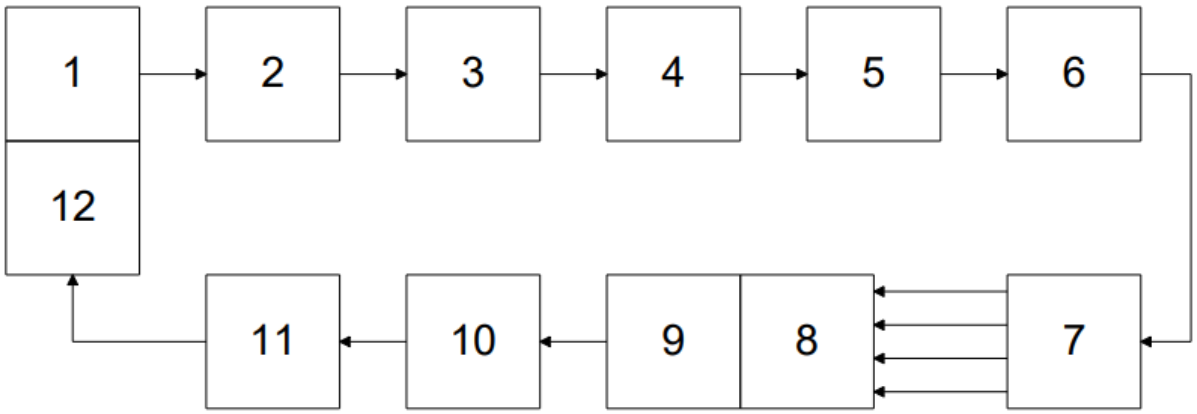

Рисунок 12. Структурна схема дистанційного керування автосигналізацією по радіоканалу.

Позначення, показані на схемі:

1-оператор дистанційного керування;

2-формувач команд управління;

3-підсилювач потужності

4-передавач

5-канал зв'язку

6-приймач

7-формувач керуючих впливів

8-об'єкт управління;

9-формувач сигналів зворотного зв'язку;

10-перетворювач;

11-канал зворотного зв'язку;

12-приймальний пристрій оператора дистанційного керування.

Розроблена структурна схема системи дистанційного керування автосигналізацією інфрачервоним каналом представлена рисунку 13.

Оператор дистанційного керування формує керуючий вплив, що надходить на формувач команд управління. Команда управління, що є електричним сигналом певної структури, надходить на підсилювач потужності, далі сигнал подається на електронно-оптичний перетворювач, де він перетворюється на інфрачервоний сигнал. Далі по інфрачервоному каналу зв'язку сигнал потрапляє на фотоелектронний перетворювач, де оптичний діапазон перетворюється на сигнал заданого виду. Підсилювач, що коригує, або послаблює сигнал до необхідної величини, далі, пройшовши регенератор і дешифратор сигнал попадає на об'єкт управління.

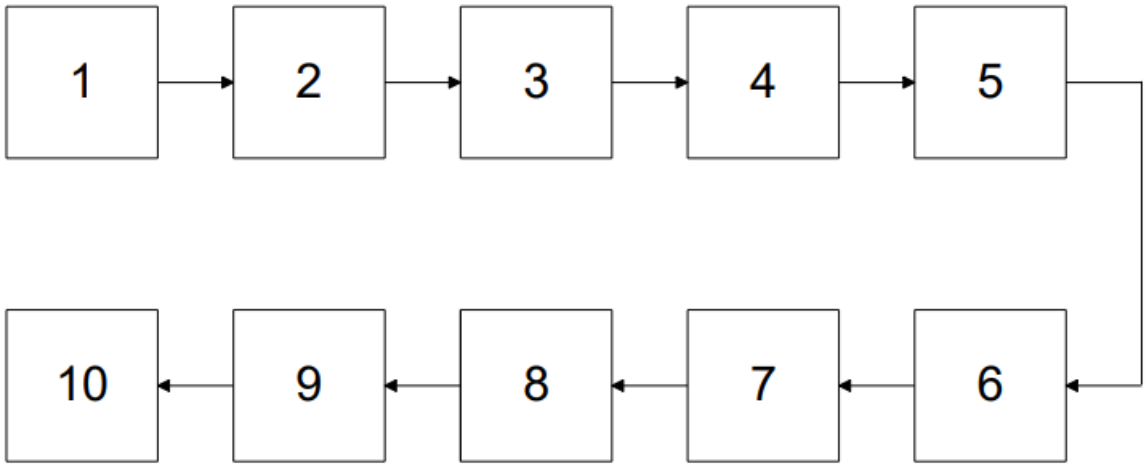

Рисунок 13. Структурна схема дистанційного керування автосигналізацією по інфрачервоному каналу.

Нижче наводяться позначення, що позначені на схемі цифрами.

1-оператор дистанційного керування;

2-формувач команд управління;

3-підсилювач потужності

4-електронно-оптичний перетворювач

5-канал зв'язку

6-фотоелектронний перетворювач

7-коригуючий підсилювач

8-регенератор

9-дешифратор

10-об'єкт управління

Розроблена структурна схема системи дистанційного керування автосигналізацією з інтегрованими радіо та інфрачервоним каналами представлена на рисунку 14.

Наступна схема є суміщенням схем автосигналізацій з використанням радіо та інфрачервоного каналу. Дане суміщення необхідне для того, щоб власник мав можливість дізнаватися, що відбувається з автомобілем, що залишився без нагляду, а також заводити машину з відстані. Всі ці сигнали повинні передаватися по радіоканалу, оскільки він забезпечує необхідну дальність передачі. Відкривати автомобіль власник може тільки по інфрачервоному каналу, який безпосередньо захищений від несанкціонованого доступу.

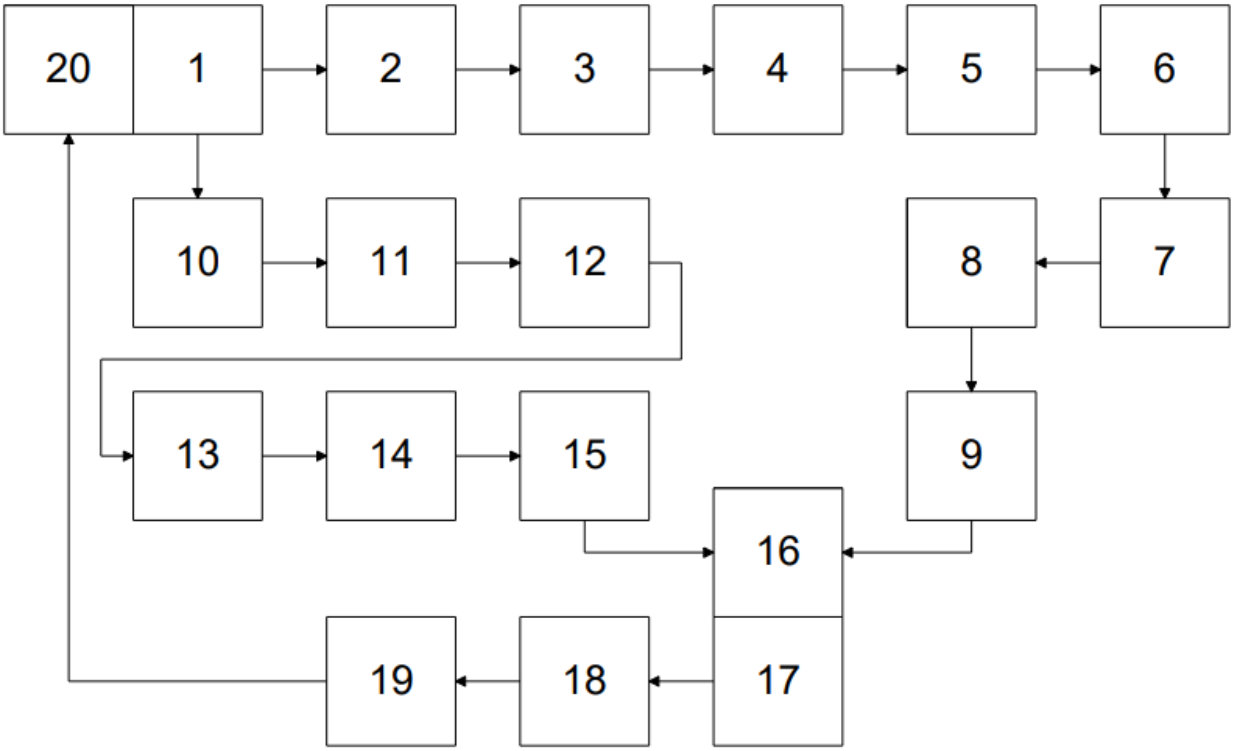

Рисунок 14. Структурна схема системи дистанційного керування автосигналізацією з інтегрованими радіо та інфрачервоним каналами.

Нижче наводяться пояснення, що позначені на схемі цифрами.

1-оператор дистанційного керування;

2-формувач команд управління;

3-підсилювач потужності

4-електронно-оптичний перетворювач

5-канал зв'язку

6-фотоелектронний перетворювач

7-коригуючий підсилювач

8-регенератор

9-дешифратор

10-формувач команд управління з радіоканалу

11- підсилювач потужності

12-передавач

13-канал

14-приймач

15- формувач керуючих впливів

16 об'єкт управління

17 - формувач сигналів зворотного зв'язку

18-перетворювач

19-канал зворотного зв'язку

20-приймальний пристрій оператора дистанційного керування.

Відповідно до наведених вище структурних схем розроблено принципові схеми формування шифруючих та дешифруючих послідовностей, передавача та приймача допоміжної системи дистанційного керування з інфрачервоним каналом зв'язку.

РОЗДІЛ 4. ПОРІВНЯЛЬНИЙ АНАЛІЗ СХЕМ ДП З ФІЗИЧНИМ І АЛГОРИТМІЧНИМ СПОСОБАМИ ФОРМУВАННЯ КЛЮЧІВ.

При створенні систем криптостійкого доступу з дистанційним управлінням перевірка шифруючих ключів на стійкість до злому є обов'язковою. Це необхідно для порівняльного аналізу псевдовипадкових послідовностей, побудованих на основі перетворень математичної логіки, і випадкових послідовностей, отриманих внаслідок оцифровки сигналів від фізичного генератора шуму.

Для перевірки можливості практичного використання такого рішення було зібрано найпростіший пристрій, що включає фізичний генератор шуму, мікроконтролер для оцифрування та розподілу послідовностей у мікросхеми пам'яті. Для оцінки статистичних властивостей підготовлених таким чином ключів, тест на частоту появи однойменних комбінацій, реалізований з використанням статистичної системи Excel показав, що шифруючі послідовності на основі фізичного шуму ближчі до абсолютно випадкового процесу, ніж послідовності сформовані методами математичної логіки (алгоритмічними). Це можна зрозуміти з результатів побудови гістограм. На фізичному генераторі – більш рівномірна (відповідає закону рівномірного розподілу послідовностей).

Отримані гістограми наведено на рисунках 15 та 16.

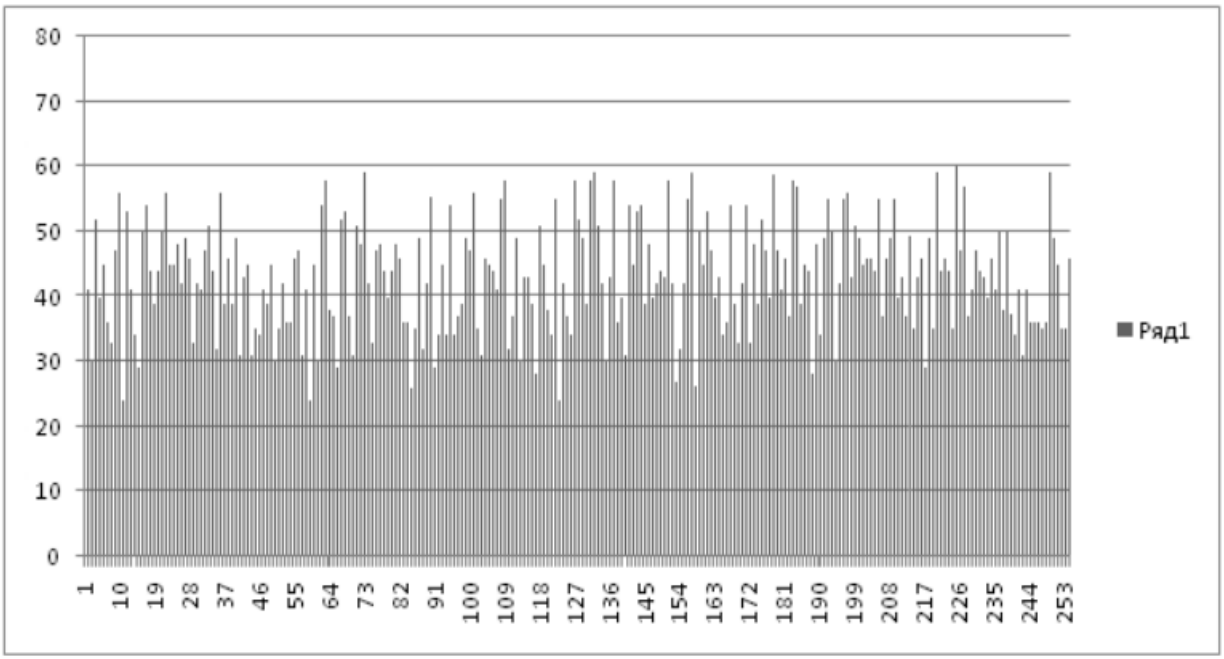

Рисунок 15. Результат перевірки послідовності псевдовипадкових чисел.

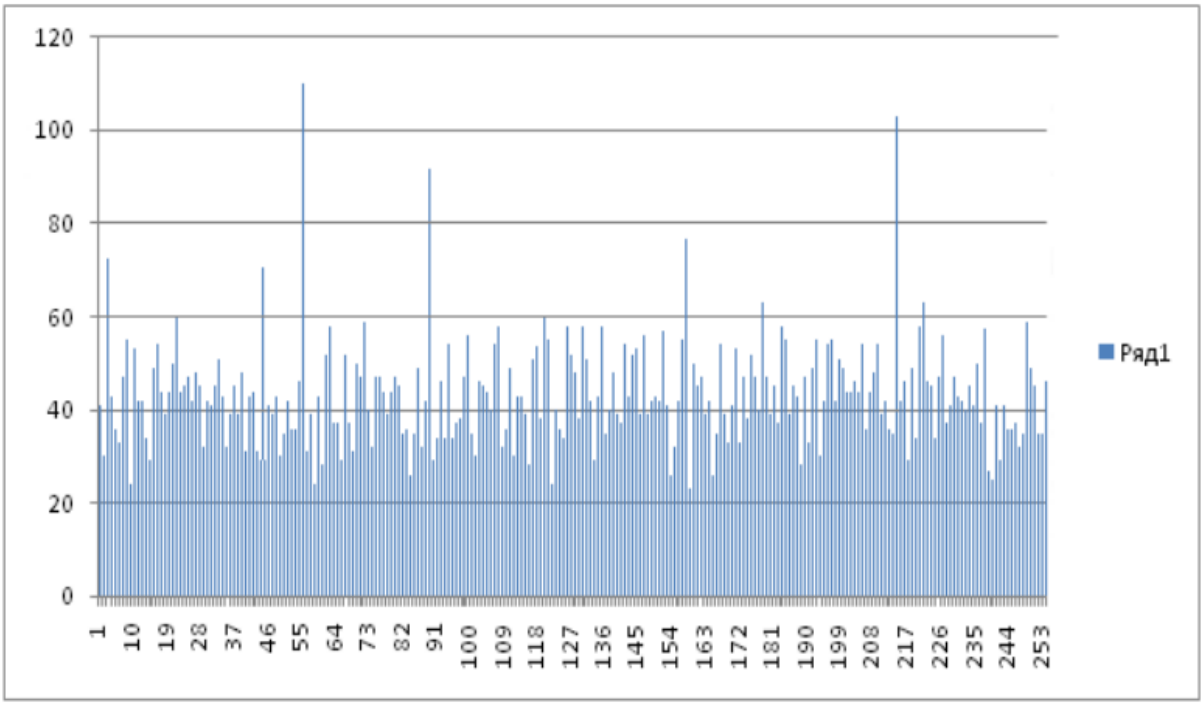

Рисунок 16. Результат перевірки послідовності псевдовипадкових чисел.

ВИСНОВКИ.

У цьому магістерському атестаційному проекті «Формування кодових структур на базі випадкових завад та криптоаналіз їх властивостей» наведено досить докладну методику для побудови додаткового методу захисту автосигналізації від злому і перехоплення сигналу.

У магістерському проекті було розроблено схему, в котрій передавач створює шуми, котрі сприяють захисту посланого сигналу від брелока до автомобіля, тим самим захищаючи від несанкціонованого доступу зловмисників.

У першому розділі докладно описані типи існуючих криптосистем і описані методи роботи ЕЦП. Другий розділ присвячений огляду принципів електронної ідентифікації. У третьому розділі вказані схеми розробленого в роботі пристрою та його складових частин. У четвертому розділі наведено порівняльний аналіз послідовностей , згенерованих внаслідок перетворень математичної логіки і випадкових послідовностей згенерованих внаслідок оцифровування сигналу з генератора шуму, який є цілком прийнятним тестом шифруючих послідовностей на криптостійкість.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.

1. Математичні основи криптографії: конспект лекцій / укладачі: В. А. Фільштінський, А. В. Бережний. - Суми: Сумський державний університет, 2011. - 138 с.
2. Steve Burnett , Stephen Paine. RSA Security's Official Guide to Cryptography 1st Edition. – New York, USA, McGraw-Hill Osborne Media; 1st edition, March 29, 2001. - 419 pages
3. В. Мао. Современная криптография: теория и практика. - СПб.: Вильямс, 2005, Д 85с.
4. Бабак В.П. Теоретичні основи захисту інформації / В. П. Бабак: Підручник. – Книжкове видавництво НАУ, 2008. – 752 с.
5. Задірака В.К. Олексик О. Комп'ютерна криптологія / В. К. Задірака, О. Олексик. – Київ, 2002. – 505 с.
6. Швець О.Ю., Лазаренко В.В. Аналіз методів і засобів захисту інформації та сучасних вимог до них / О.Ю. Швець, В.В. Лазаренко
7. Гайворонський М.В. Безпека інформаційно-комунікаційних систем / М.В. Гайворонський, О.М. Новіков. – К.: Видавнича група BHV, 2009. – 608 с.
8. Дронь М.М. Основи теорії захисту інформації: Навч. посібник / М.М. Дронь, В.П. Малайчук, О.М. Петренко. – Д.: Вид-во Дніпропетр. ун-ту, 2001. – 312 с.
9. Захист інформації – українське законодавство у сфері захисту інформації [Електронний ресурс]. – Режим доступу: http://bit.ly/2slbtSP. – Назва з екрану.
10. Конахович Г.Ф. Захист інформації в мережах передачі даних: Підручник / Г.Ф. Конахович, О.Г. Корченко, О.К. Юдін. – К.: Видавництво ТОВ НВП «ІНТЕРСЕРВІС», 2009. – 714 с.
11. Вишня В. Б. Основи інформаційної безпеки : Навч. посіб. / В.Б. Вишня, О.С. Гавриш, Е.В. Рижков. – Дніпро : ДДУВС, 2020. – 128 с.

12. Kahn, David. The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. – London, Macmillan Publishers, 1997
13. Горбенко І. Д. Захист інформації в інформаційно-телекомунікаційних системах : навч. посібник. Частина 1 : Криптографічний захист інформації / І. Д. Горбенко, Т. О. Гріненко. – Харків : ХНУРЕ, 2004. – 368 с.
14. Friedrich L. Bauer, Decrypted Secrets: Methods and Maxims of Cryptology,- New York, Springer-Verlag New York Inc (C), February 1, 2000. – 470 pages
15. Кнут Д. Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд. : Пер. с англ. : Уч. пос. / Дональд Эрвин Кнут. – М.: Издательский дом Вильямс, 2010. – 720 с. : ил. — Парал. тит. англ.

ДОДАТКИ

Додаток1

```
'-----------------------------------------------------------------------
'генерує таблицю випадкових чисел для IR сигналізації і записує
'їх в зовнішньому EEPROM AT24C64 "SOURCE" (для передавача)
'для копіювання EEPROM для приймача використовуємо EEPROM AT24C64 "COPY"
'генератор - на основі шуму стабілітрона с наступною оцифровкою АЦП
' Controller : ATMega88V-10AU
' Compiler : BASCOM-AVR Rev. 1.11.9.1
'-----------------------------------------------------------------------
'
$regfile = "m88def.dat"
$crystal = 8000000 'внутрішній RC генератор на 8мГц або зовнішній кварц
$baud = 19200
'********** налаштування АЦП
******************************************
' використовуємо внутрішнє джерело опорної напруги Vref=1,1b
' ADC_1=Vin*1024/Vref
' де Vin - с виходу генератора шуму (вхід ADC1)
```

```
Config Adc = Single , Prescaler = Auto , Reference = Internal_1.1
' Avcc
' Internal_1.1
Start Adc
Dim Adc_1 As Word , Channel As Byte
Channel = 1
'приклад читання АЦП:
' ADC_1 = Getadc(channel)
'*****************************************************************************
*******
'******************** шина I2C
*********************************************
'конфігуруємо шину I2C для зовнішніх EEPROM AT24C64 "SOURCE" і "COPY"
Config Sda = Portb.1
Config Scl = Portb.2
' підпрограми для EEPROM "SOURCE"
Declare Sub Write_eep_source(byval E_adr As Word , Byval E_dat As Byte)
Declare Sub Read_eep_source(byval E_adr As Word , E_dat As Byte)
Dim Source_dat As Byte
' дані EEPROM "SOURCE"
' підпрограми для EEPROM "COPY"
Declare Sub Write_eep_copy(byval E_adr As Word , Byval E_dat As Byte)
```

```
Declare Sub Read_eep_copy(byval E_adr As Word , E_dat As Byte)
Dim Copy_dat As Byte
' дані EEPROM "COPY"
Dim Epp_cnt As Word
Dim E_adr As Word
' адреса EEPROM AT24C64
Dim E_dat As Byte
' дані запису/читання EEPROM AT24C64
Dim E_adr_h As Byte
' старший байт адреси EEPROM AT24C64
Dim E_adr_l As Byte
' молодший байт адреси EEPROM AT24C64
Const Eep_source_wr = &B10100000
' адреса EEPROM "SOURCE" для запису
Const Eep_source_rd = &B10100001
' адреса EEPROM "SOURCE" для читання
Const Eep_copy_wr = &B10100010
' адреса EEPROM "COPY" для запису
Const Eep_copy_rd = &B10100011
' адреса EEPROM "COPY" для читання
```

```
'*****************************************************************************
*******

' підпрограми генерації байта випадкового числа

Declare Sub Gen_rnd(random As Byte)

Dim Random As Byte , Bit_cnt As Byte , Zero_bit As Byte

Dim Tmp_dat As Byte

'***************************** світлодіоди

***********************************************

Config Portd.5 = Output

Gen_led Alias Portd.5 'світлодіод "генерація" GEN_LED=1 : світлодіод
погашений

Config Portd.6 = Output

Copy_led Alias Portd.6  'світлодіод "копіювання" COPY_LED=1 : світлодіод
погашений

'*****************************************************************************
********* **************

'****************************** кнопки

************************************

'на вхід INT0 поданий сигнал з кнопки "GEN&PROG"

' (старт генерації випадкових чисел і запис їх в EEPROM "SOURCE")

Config Pind.2 = Input

Set Portd.2 ' pull-up resistor
```

```
Config Int0 = Low Level

On Int0 Gen_tab Nosave

'на вхід INT1 поданий сигнал з кнопки "COPY"

' (копіювання з EEPROM "SOURCE" в EEPROM "COPY"

Config Pind.3 = Input

Set Portd.3 ' pull-up resistor

Config Int1 = Low Level

On Int1 Copy_tab Nosave
'*************************************************************************
*******' '////////////////// interrupts //////////////////////

Enable Int0 ' кнопка "GEN&PROG"

Enable Int1 ' кнопка "COPY" Enable Interrupts

'////////////////////////////////////////////////

' глобальні змінні

Dim Dev_mode As Byte ' режим роботи

' DEV_MODE=0 - фоновий режим, генерація шуму и вивід його на RS232 для
тестування

' DEV_MODE=1 - генерація таблиць і запис їх в EEPROM "SOURCE"

' DEV_MODE=2 - режим копіювання таблиць з EEPROM "SOURCE" в EEPROM
"COPY"

' ++++++++++ вихідний стан +++++++++++++++++

' погасити світлодіоди
```

```
Gen_led = 1

Copy_led = 1

Dev_mode = 0

'---[ main program loop ]-------------------------------------------------

Do

Select Case Dev_mode

Case 0:

' фонова генерація випадкового числа для відладки

Call Gen_rnd(random)

Print Random

Gen_led = 0

Waitms 70

Gen_led = 1

Waitms 70

Case 1:

' тут буде генерація таблиць и запис їх в EEPROM "SOURCE" ' і в EEPROM
"COPY"

Gen_led = 0

Print "Generation start"

For E_adr = 0 To 8191

        Call Gen_rnd(random)
```

```
        ' запис в EPP "SOURCE"
        Call Write_eep_source(e_adr , Random)
        Call Read_eep_source(e_adr , E_dat)
        Print "ADR=" ; E_adr ; "; RANDOM=" ; Random ; "; E_dat=" ; E_dat ' для
отладки
        ' запис в EPP "COPY"
        Call Write_eep_copy(e_adr , Random)
        Waitms 100
Next E_adr
Dev_mode = 0
Gen_led = 1
Print "Generation stop"
Case 2:
'порівняння таблиць Eeprom "SOURCE" і Eeprom "COPY"
Copy_led = 0
For E_adr = 0 To 8191
        Call Read_eep_source(e_adr , Source_dat) ' читаемо "SOURCE"
        Call Read_eep_copy(e_adr , Copy_dat) ' читаемо "COPY"
        Print "ADR=" ; E_adr ; "; SOURCE=" ; Source_dat ; "; COPY=" ; Copy_dat;
If Source_dat = Copy_dat Then
Print "/ EQUAL"
```

```
Else Print "/ not equal !"
End If
Waitms 100
Next E_adr
Dev_mode = 0
Copy_led = 1
Case Else:
End Select
Loop
' ---------------------- ПІДПРОГРАМИ -----------------------------------
' ===== підпрограма обробки interrupt по входу INT0 (кнопка "GEN&PROG")
====
Gen_tab:
Dev_mode = 1 ' дозволити режим генерації випадкових чисел и запис їх в
EEPROM "SOURCE"
Disable Int0
Waitms 300 'проти тремтіння контакту
' чекаємо, поки кнопку відпустять:
Do
 Loop Until Pind.2 = 1
Return
```

```
'=============================================================
======= ============
' ======= підпрограма обробки interrupt по входу INT1 (кнопка "COPY")
======
Copy_tab:
Disable Int1
Dev_mode = 2 ' дозволити режим копіювання
Waitms 300 'проти тремтіння контакту
' чекаємо, поки кнопку відпустять:
Do
Loop Until Pind.3 = 1
Return
'=============================================================
======= ============
'%%%%%%%%%%% підпрограми запису/читання EEPROM
"SOURCE" %%%%%%%%%%%%%%%%%%%%%
'підпрограма запису байта в EEPROM "SOURCE"
Sub Write_eep_source(byval E_adr As Word , Byval E_dat As Byte)
E_adr_h = High(e_adr) 'старший байт адреси
E_adr_l = Low(e_adr) 'молодший байт адреси
I2cstart 'старт
I2cwbyte Eep_source_wr 'адреса чипа для запису
```

```
I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C
I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C
I2cwbyte E_dat 'передати дані E_dat для запису на шину I2C
I2cstop 'стоп
Waitms 10 'затримка 10 ms
End Sub
'підпрограма читання байта з EEPROM "SOURCE"
Sub
Read_eep_source(byval E_adr As Word , E_dat As Byte)
E_adr_h = High(e_adr) 'старший байт адреси
E_adr_l = Low(e_adr) 'молодший байт адреси
I2cstart 'старт
I2cwbyte Eep_source_wr 'адреса чипа для запису
I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C
I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C
I2cstart 'повторити старт
I2cwbyte Eep_source_rd 'адреса чипа для читання
I2crbyte E_dat , Nack 'прочитати байт E_dat з шини I2C
I2cstop 'стоп
```

```
End Sub
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'########### підпрограми запису/читання EEPROM "COPY"
######################
'підпрограми запису байта в EEPROM "COPY"
Sub Write_eep_copy(byval E_adr As Word , Byval E_dat As Byte)
E_adr_h = High(e_adr) 'старший байт адреси
E_adr_l = Low(e_adr) 'молодший байт адреси
I2cstart 'старт
I2cwbyte Eep_copy_wr 'адрес чипа для запису
I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C
I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C
I2cwbyte E_dat 'передати дані E_dat для запису на шину I2C
I2cstop 'стоп
Waitms 10 'затримка 10 ms
End Sub
'підпрограма читання байту з EEPROM "COPY"
Sub Read_eep_copy(byval E_adr As Word , E_dat As Byte)
E_adr_h = High(e_adr) 'старший байт адреси
E_adr_l = Low(e_adr) 'молодший байт адреси
I2cstart 'старт
```

```
I2cwbyte Eep_copy_wr 'адреса чипа для запису
I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C
I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C
I2cstart 'повторити старт
I2cwbyte Eep_copy_rd 'адреса чипу для читання
I2crbyte E_dat , Nack 'прочитати байт E_dat з шини I2C
I2cstop 'стоп
End Sub
'#########################################################################
######### ##
' підпрограма генерації байту випадкового числа
' використати мол. біт АЦП
Sub Gen_rnd(random As Byte)
Random = 0
Bit_cnt = 0
For Bit_cnt = 0 To 7
        Adc_1 = Getadc(1)
        Zero_bit = Low(adc_1)
        Zero_bit = Zero_bit And 1
        Shift Random , Left Random = Random + Zero_bit
        Next Bit_cnt
```

```
End Sub
' End
```

Додаток 2.

```
'-------------------------------------------------------------------------------
' Filename : IR_TX_LOCK_v4.bas
' counter натискань
' таблиця ключів - в зовнішньому EEPROM AT24C64
'призначення: передатчик IR сигналізації
' прототип : Ger langezaal (AN105)
' Controller : ATMega48PA-AU
' Compiler : BASCOM-AVR Rev. 1.11.9.5
'-------------------------------------------------------------------------------
'
$eepleave 'not to recreate or erase the EEP file
$regfile = "m48def.dat"
$crystal = 8000000 'зовнішній кварц 8мГц
$baud = 19200
' АЦП********** додано 07.02.2010 ********************************
' Налаштування АЦП
' використовуємо джерело опорної напруги 1,1в
' ADC_1=Vin*1024/Vref
```

```
' де Vin=Vcc/3 (на вхід ADC1 через резистивний поділювач R6-R7
' подано 1/3 от напруги батареї Vcc) , Vref=1,1v
'в результаті виходить ADC_1= Vcc*1024/3,3
' допустима напруга живлення для ATMega88V 1,8в
' при Vcc=1,8v отримуємо ADC_1= 558. Це число будемо контролювати
'при кожному натисканні кнопки. Результат супроводжувати звуком.
 'Config Adc = Single , Prescaler = Auto , Reference = Internal_1.1
 'Start Adc
 'Dim Adc_1 As Word , Channel As Byte
 'Channel = 1
'приклад читання АЦП:
' ADC_1 = Getadc(channel)
'
**********************************************************************
******
'конфігуруємо шину I2C для зовнішнього EEPROM AT24C64
Config Scl = Portb.3
Config Sda = Portb.2
Declare Sub Write_eeprom(byval E_adr As Word , Byval E_dat As Byte)
Declare Sub Read_eeprom(byval E_adr As Word , E_dat As Byte)
Dim E_adr As Word 'адреса EEPROM AT24C64
```

```
Dim E_dat As Byte ' данні запису/читання EEPROM AT24C64
Dim E_adr_h As Byte ' старший байт адреси EEPROM AT24C64
Dim E_adr_l As Byte ' молодший байт адреси EEPROM AT24C64
Const Eeprom_write = &B10100000 ' адреса чипа AT24C64 для запису
Const Eeprom_read = &B10100001 ' адреса чипа AT24C64 для читання
'********** ZOOMER
**************************************************
Speaker Alias Portd.5 'зуммер (+)
Speaker1 Alias Portd.6 'зуммер (-)
Config Speaker = Output
Config Speaker1 = Output
'
**********************************************************************
******
'EEPROM використовується для запису і зберігання counter натискань Tx_count
'ячейку с адресою 0 не використовуємо
Dim Tx_cnt_epp As Eram Word At 1 ' counter передатчика в EEPROM
Dim Tx_cnt As Word ' counter передатчика в ОЗУ
Dim I As Byte 'counter циклу
Dim Tx_buf(9) As Byte ' послідовність - 9 байт
' передаються байти:
'Tx_buf(1)= &HCA - заголовок (фіксоване число)
```

```
'Tx_buf(2) - команда:
' Tx_buf(2)= &h 00 - тест
' Tx_buf(2)=&h 01 - закрити
' Tx_buf(2)=&h 02 - відкрити
'Tx_buf(3)= counter натиснутий (старший байт Tx_count)
'Tx_buf(4)= counter натиснутий (молодший байт Tx_count)
'Tx_buf(5)= EEP_EXT(Tx_count*4)
'Tx_buf(6)= EEP_EXT(Tx_count*4+1)
'Tx_buf(7)= EEP_EXT(Tx_count*4+2)
'Tx_buf(8)= EEP_EXT(Tx_count*4+3)
''Tx_buf(9)=CRC8 - контрольна сума
Dim Tx_byte As Byte
Dim Bit_num As Byte
Dim Cmd As Byte
 ' Cmd = 1 : закрити
 ' Cmd = 2 : відкрити
 ' Cmd = 3 : тест
Dim Cnt_16 As Word
'********** світлодіоди *********
Config Portc.1 = Output
Sync Alias Portc.1 'світлодіод передатчика або синхр. осцилографа
```

```
Config Portc.2 = Output
Mode_led Alias Portc.2 'світлодіод режима роботи (Mode_led=1 : світлодіод погашений)
'************************************
' 36kHz carrier reload value
 'Const T_oc1 = 55 'value for 4MHz crystal
'Const T_oc1 = 110 'value for 8MHz crystal
 'Const T_oc1 = 138 'value for 10MHz crystal
' 33kHz carrier reload value
Const T_oc1 = 120 'value for 8MHz crystal
Const Carrier_on = &B01000000 'IR 36kHz carrier on , передача "1"
'Const Carrier_off = &B11000000 'IR 36kHz carrier off , якщо потрібна "1" при передачі "0"
Const Carrier_off = &B10000000 'IR 36kHz carrier off , если потрібен "0" при передачі "0"
Config Timer1 = Timer , Prescale = 1 , Compare A = Toggle , Clear Timer = 1
Timer1 = 0
Compare1a = T_oc1 'Pb.1 = OC1A = IR carrier output 36 kHz
Declare Sub Logic_0
Declare Sub Logic_1
Tccr1a = Carrier_off
```

```
'на вхід INT0 поданий сигнал з кнопки ЗАКРИТИ
Config Pind.2 = Input
Set Portd.2 ' pull-up resistor
Config Int0 = Low Level 'на INT0 підключена кнопка ЗАКРИТИ
On Int0 On_lock Nosave
'на вхід INT1 поданий сигнал з кнопки ВІДКРИТИ
Config Pind.3 = Input
Set Portd.3 ' pull-up resistor
Config Int1 = Low Level 'на INT1 підключена кнопка ВІДКРИТИ
On Int1 Off_lock Nosave
Enable Interrupts
' погасити світлодіоди
Sync = 1
Mode_led = 1
'---[ main program loop ]-------------------------------------------------
69
Do
'дозволимо прокидання при натисканні кнопок
Enable Int0
Enable Int1
' і переходимо у сплячий режим
```

```
'Idle

'Powersave

Powerdown

Waitms 20 ' чекаємо 20мс після прокидання

Sync = 0 'sync<--0 для відладки (синхр. осцилографа)

Ddrb.1 = 1 'set OC1A (Pb.1) = IR carrier output

'сформуємо послідовність TX_BUF

Tx_cnt = Tx_cnt_epp ' прочитати counter из EEPROM

Tx_buf(1) = &HCA ' ідентифікатор для синхронізації приймача

Tx_buf(2) = Cmd ' команда 1- закрити, 2 - відкрити, 0 - тест

Tx_buf(3) = High(tx_cnt) 'ст. байт tx_count

Tx_buf(4) = Low(tx_cnt) 'мол. байт tx_count

E_adr = Tx_cnt * 4 ' передаемо 4 байти на одне значення Tx_cnt

Call Read_eeprom(e_adr , Tx_buf(5))

E_adr = E_adr + 1

Call Read_eeprom(e_adr , Tx_buf(6))

E_adr = E_adr + 1

Call Read_eeprom(e_adr , Tx_buf(7))

E_adr = E_adr + 1

Call Read_eeprom(e_adr , Tx_buf(8))

Tx_buf(9) = Crc8(tx_buf(1) , 8) 'контрольна сума
```

```
For I = 1 To 9
 For Bit_num = 7 To 0 Step -1 '8 bit , msb first (ст. біт вперед)
70
 If Tx_buf(i).bit_num = 0 Then Logic_0 Else Logic_1
 Next
Next
Sync = 1 'sync<--1 для відладки
 Tccr1a = Carrier_off
'************************ для відладки вивід в COM порт ***************
 For I = 1 To 9
 Print Hex(tx_buf(i)) ; " ";
 Next
 Print
'*************************************************************************
*********
 Waitms 200 'frame gap delay
Tx_cnt = Tx_cnt + 1
' Зовнішнє EPPROM AT24C64 місткістю 8192 байта (2048 записів по 4 байта)
If Tx_cnt = 2048 Then Tx_cnt = 0
Tx_cnt_epp = Tx_cnt ' зберегти counter в EEPROM
' ****************************************
```

```
' перевірити напругу живлення
'Adc_1 = Getadc(channel)
'If Adc_1 < 558 Then
Loop
'-------------------------------------------------------------------------
Sub Logic_0
 Tccr1a = Carrier_on
 Waitus 883
 Tccr1a = Carrier_off
 Waitus 884
End Sub
Sub Logic_1
 Tccr1a = Carrier_off
 Waitus 883
 Tccr1a = Carrier_on
 Waitus 884
End Sub
'=========================================================
' підпрограма обробки interrupt по входу INT0 (кнопка ЗАКРИТИ)
On_lock:
Disable Int0
```

```
Cmd = 1
'Tx_buf(2) = &H01
Set Speaker
Reset Speaker1
'Waitms 300 'проти тремтіння контакту
' чекаємо, коли кнопку відпустять:
Do
Toggle Speaker
Toggle Speaker1
Waitus 125
Loop Until Pind.2 = 1
Reset Speaker
Reset Speaker1
End_int0:
Return
'=========================================================
' підпрограма обробки interrupt по входу INT1 (кнопка ВІДКРИТИ)
Off_lock:
Disable Int1
Cmd = 2
'Tx_buf(2) = &H02
```

```
Set Speaker
Reset Speaker1
'Waitms 300 'проти тремтіння контакту
' чекаємо, поки кнопку відпустять:
Do
Toggle Speaker
Toggle Speaker1
Waitus 125
Loop Until Pind.3 = 1
Reset Speaker
Reset Speaker1
Return
'=======================================================
'підпрограма читання байта з EEPROM AT24C64
Sub Read_eeprom(byval E_adr As Word , E_dat As Byte)
 E_adr_h = High(e_adr) 'старший байт адреси
 E_adr_l = Low(e_adr) 'молодший байт адреси
 I2cstart 'старт
 I2cwbyte Eeprom_write 'адреса чіпу для запису
 I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C
 I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C
```

```
 I2cstart 'повторити старт
 I2cwbyte Eeprom_read 'адреса чіпа для читання
 I2crbyte E_dat , Nack 'прочитати байт E_dat з шини I2C
I2cstop 'стоп
End Sub
```

Додаток 3.

```
'-------------------------------------------------------------------------------
' Filename : IR_RX_LOCK_V2.bas
' призначення : IR приймач
'таблиці з ключами - в зовнішній EPPROM AT24C64 (8192 байт)
'один запис - 4 байти
' Controller : ATMega88V-10AU
' Compiler : BASCOM-AVR Rev. 1.11.9.1
'-------------------------------------------------------------------------------
'
$eepleave 'not to recreate or erase the EEP file
$regfile = "m88def.dat" 'the chip type, subtitutes with the one you like
$crystal = 8000000
$baud = 19200
```

```
'
**********************************************************************
******
'конфігуруємо шину I2C для зовнішнього EEPROM AT24C64
Config Scl = Portc.3
Config Sda = Portc.4
Declare Sub Read_eeprom(byval E_adr As Word , E_dat As Byte)
Dim E_adr As Word 'адреса EEPROM AT24C64
Dim E_dat As Byte ' дані запису/читання EEPROM AT24C64
Dim E_adr_h As Byte ' старший байт адреси EEPROM AT24C64
Dim E_adr_l As Byte ' молодший байт адреси EEPROM AT24C64
Const Eeprom_write = &B10100010 ' адреса чипа AT24C64 для запису
(ADR=001)
Const Eeprom_read = &B10100011 ' адреса чипа AT24C64 для читання
(ADR=001)
'
**********************************************************************
******
'EEPROM процесора використовуємо для запису і збереження counter-а
натискань
'і режиму роботи
```

75

```
'комірку с адресою 0 не використовуємо
```

```
Dim Rx_cnt_used As Eram Word At 1 'використане значення counter-а адреси таблиці в
EEP
Dim Tmp_word As Word ' часова змінна типу word
Const Rx_cnt_max = 2047 'максимальна кількість записів 2048 в зовнішньому EEPROM
Dim Rx_mode As Eram Byte At 3 'режим роботи (0- очікування, 1- охорона)
Dim Rx_cnt As Word ' counter приймача в ОЗУ
Dim Tmp As Byte 'часова змінна
' Dim Start_time As Byte 'для відладки
Dim Rx_buf(9) As Byte ' прийнята послідовність
' приняті байти:
'Rx_buf(1)= &CA - ідентифікатор (фіксоване число)
'Rx_buf(2) - команда:
' Rx_buf(2)= &h 00 - тест
' Rx_buf(2)=&h 01 - закрити
' Rx_buf(2)=&h 02 - відкрити
'Rx_buf(3)= counter натискань (старший байт Rx_cnt)
'Rx_buf(4)= counter натискань (молодший байт Rx_cnt)
'Rx_buf(5)= EEPROM_EXT(Rx_cnt*8)
'Rx_buf(6)= EEPROM_EXT(Rx_cnt*8+1)
```

```
'Rx_buf(7)= EEPROM_EXT(Rx_cnt*8+2)
'Rx_buf(8)= EEPROM_EXT(Rx_cnt*8+3)
'Rx_buf(9)=CRC8 - контрольна сума
'TIMER0 для вимірювання часу. 1тік=32мкс
Config Timer0 = Timer , Prescale = 256
'часові затримки, контрольовані Timer0 (в тіках)
Const T0 = 1777 'мкс – час записування одного біту
Const Ts1 = 22 ' TS1=T0/2-20%=888*0.8=710мкс = 710/32= 22 тік -
мінімальна продовжуваність старту
Const Ts2 = 33 ' TS2=T0/2+20%=888*1.2=1066мкс = 1066/32= 33 тік -
максимальна продовжуваність старту
Const Tnext = 1333 ' Tnext= 1333мкс - затримка для зчитування біту після
фронту
Dim Lastbit As Bit ' минуле значення сигналу для визначення фронту
Dim Bitcnt As Byte ' counter прийнятих біт
Dim Rx_byte As Byte ' прийнятий байт
Dim Bit_num As Byte 'номер біту
Dim Rx_byte_num As Byte 'номер байту
Dim Rx_status As Byte ' стан приймача
' Rx_status=0 - прийом послідовності
' Rx_status=1 - приняті всі байти без помилок
```

```
' Rx_status=2 - помилка старту

' Rx_status=3 - помилка прийому посилань

' Rx_status=4 - помилка контрольної суми

Stop Timer0 'stop timer0 first for other process

'виходи на виконавче реле

Config Portd.5 = Output ' управління реле 842-1C-C

Rele Alias Portd.5 'rele=1 -реле вмк.

Config Portb.0 = Output ' вмк поляр. реле РПС20

P_rele_on Alias Portb.0

Config Portd.7 = Output 'вимк поляр. реле РПС20

P_rele_off Alias Portd.7

Config Portd.6 = Output 'світлодіод

Led Alias Portd.6

Config Pind.3 = Input 'вхід від IR приймача (INT1)

Set Portd.3 'set the pull up resistor

Pin_ir Alias Pind.3 'alias for easy naming

Config Int1 = Low Level 'configure int1 on low level

On Int1 Isr_int1 'the isr label

Enable Interrupts

'для перевірки світлодіода вимк-ввімк 2 раза при вмк. живлення:

Led = 1 'вмк. світлодіод
```

```
Waitms 100
Led = 0 'вимк. світлодіод
Waitms 300
Led = 1 'вмк. світлодіод
Waitms 100
Led = 0 'вимк. світлодіод
' для відладки
Print "IR RX LOCK Ver 2.0 20.03.2011"
Rx_cnt = Rx_cnt_used
Print "RX_CNT=" ; Rx_cnt
If Rx_cnt_used = &HFFFF Then Rx_cnt_used = 0 'початкова ініціаліз. Counter-a
'відновити стан звичайного реле, якщо вимикали живлення
If Rx_mode = 1 Then
Rele = 1
Else
Rele = 0
End If
'---[ main program loop ]-------------------------------------------------------
Do
P_rele_on = 0
P_rele_off = 0
```

```
Rx_status = 0
'очистити буфер
For Rx_byte_num = 1 To 9
Rx_buf(rx_byte_num) = 0
Next Rx_byte_num
Enable Int1
Rx_wait:
' в режимі охорони світлодіод блимає
If Rx_mode = 1 Then
Led = 1
Waitms 200
Led = 0
Waitms 800
End If
Select Case Rx_status
Case 0 : Goto Rx_wait 'нічого не прийнято
Case 2 To 4 : Print "Error=" ; Rx_status 'помилка
Case 1: 'прийняті пакети без помилок
' для відладки виведемо на термінал
 For Tmp = 1 To 9
 Print Hex(rx_buf(tmp)) ; " ";
```

```
 Next
Print Start_time
' прочитаємо counter
Rx_cnt = Rx_buf(3) * 256
Rx_cnt = Rx_cnt + Rx_buf(4)
Tmp_word = Rx_cnt_used 'прочитаємо старе значення counter з EEP
' і звіримо зі старим значенням (блокуємо команди, раніше відправлені в ефір)
If Rx_cnt > Tmp_word Then Goto Rx_ok 'перевірка на "свіжість"
'перевіримо крайове значення, коли Rx_cnt=0 (counter пішов по наступному колу)
If Rx_cnt <> 0 Then Goto Err_cnt
If Tmp_word <> Rx_cnt_max Then Goto Err_cnt
'порівняємо прийняті дані з таблицею
'зберігаємо новий counter в EEP
Rx_ok:
Rx_cnt_used = Rx_cnt
E_adr = Rx_cnt * 4 ' передаємо 4 байти на одне значення Tx_cnt
Call Read_eeprom(e_adr , Tmp)
If Tmp <> Rx_buf(5) Then Goto Err_tab
E_adr = E_adr + 1
Call Read_eeprom(e_adr , Tmp)
```

```
If Tmp <> Rx_buf(6) Then Goto Err_tab
E_adr = E_adr + 1
Call Read_eeprom(e_adr , Tmp)
If Tmp <> Rx_buf(7) Then Goto Err_tab
E_adr = E_adr + 1
Call Read_eeprom(e_adr , Tmp)
If Tmp <> Rx_buf(8) Then Goto Err_tab
' Rx_buf(2)= &h 00 - тест
' Rx_buf(2)=&h 01 - закрити
' Rx_buf(2)=&h 02 - відкрити
 Select Case Rx_buf(2)
 Case 0: 'test
 Case 1 :
Rx_mode = 1 'вмк. режим охорони
 Rele = 1 'вмк звичайне реле
 P_rele_off = 1 ' подати імпульс закриття пол.реле
 Waitms 100
 P_rele_off = 0 ' закінчити імпульс закриття пол.реле
 ' Waitms 100
 Case 2 :
 Rx_mode = 0 'зняти охорону
```

```
 Rele = 0 'вимк звичайне реле
 P_rele_on = 1 ' подати імпульс відкриття пол.реле
 Waitms 100
 P_rele_on = 0 ' закінчити імпульс відкриття пол.реле
 ' Waitms 100
 Case Else :
 End Select
Err_tab: 'не співпала з таблицею
Err_cnt: ' стара команда
Case Else : Print "not used"
End Select
Waitms 100 ' чекаємо 1 сек перед прийомом наступної команди
Loop
'=========================================================
' підпрограма обробки зупинки по входу INT1 (IR приймач)
Isr_int1:
Disable Int1 'зупинка тільки від стартового імпульсу
' виміряємо продовжуваність стартового імпульсу T_syn_pulse_0 таймером Timer0
' Номінально T_syn_pulse_0 = T0/2= 888 мкс = 888/32= 28 тик ( prescale=256)
' вимірюється в межах Ts1<T_syn_pulse_0<Ts2 (22 - 33 тик)
```

```
 Timer0 = 0
 Start Timer0
 Bitwait Pin_ir , Set 'чекаємо, коли вхід IR стане "1"
 Stop Timer0 'stop timer0
 'перевіряємо стартовий біт на допуск - від TS1 до TS2
 Tmp = Timer0
Start_time = Timer0
If Tmp > Ts1 And Tmp < Ts2 Then
Goto Sync_ok
Else
Rx_status = 2 ' помилка старту
Goto Re_turn
End If
' прийнятий стартовий біт, починаємо декодувати
Sync_ok:
Bitcnt = 1 'один біт (стартовий) ми вже прийняли, це Rx_buf(1).7=1
Rx_buf(1) = &B10000000 ' стартовий біт=1 - старший
Rx_byte_num = 1 'номер байту в Rx_buf()
Bit_num = 6 ' починаємо прийом з Rx_buf(1).6
Next_bit:
Lastbit = Pin_ir
```

```
Timer0 = 0
Start Timer0
Wait_edge:
If Timer0 > Ts2 Then
 Stop Timer0
 Rx_status = 3 ' помилка прийому пакетів
Goto Re_turn
Else
 If Lastbit = Pin_ir Then Goto Wait_edge
End If
Stop Timer0
' можна читати біт
' прийняти Pin_ir в буфер з інверсією
 If Pin_ir = 0 Then Rx_buf(rx_byte_num).bit_num = 1 ' інверсія
Waitus Tnext ' Tnext=3/4T0 = 1333мкс
' counter циклу по бітам
If Bit_num = 0 Then
'прийнятий байт, готуємось до прийому наступного
Bit_num = 7
Rx_byte_num = Rx_byte_num + 1
Else
```

```
Bit_num = Bit_num - 1 ' наступний біт

End If

Bitcnt = Bitcnt + 1

If Bitcnt < 72 Then Goto Next_bit

'приняті всі біти, перевіримо контрольну суму

If Rx_buf(9) <> Crc8(rx_buf(1) , 8) Then

'виявлена помилка CRC

Rx_status = 4 ' помилка прийому посилань

Goto Re_turn

Else

Rx_status = 1 ' прийняті всі байти без помилок

End If

Re_turn:

Return

'=====================================================

'підпрограма читання байту з EEPROM AT24C64

Sub Read_eeprom(byval E_adr As Word , E_dat As Byte)

 E_adr_h = High(e_adr) 'старший байт адреси

 E_adr_l = Low(e_adr) 'молодший байт адреси

 I2cstart 'старт

 I2cwbyte Eeprom_write 'адреса чіпа для запису
```

```
 I2cwbyte E_adr_h 'передати старший байт адреси на шину I2C

 I2cwbyte E_adr_l 'передати молодший байт адреси на шину I2C

 I2cstart 'повторити старт

 I2cwbyte Eeprom_read 'адреса чіпа для читання

 I2crbyte E_dat , Nack 'прочитать байт E_dat з шини I2C

 I2cstop 'стоп

End Sub

'=======================================================
```