

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

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(назва факультету)

Кафедра кібербезпеки

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(освітній рівень)

на тему: "Розробка і налаштування системи сигналізації з
дистанційним керуванням"

Виконав: студент (ка)

Спеціальності:

125 «Кібербезпека»

(шифр і назва напрямку підготовки, спеціальності)

Галанський І.О.

підпис

(прізвище та ініціали)

Керівник

Лечаченко Т.А.

підпис

(прізвище та ініціали)

Нормоконтроль

Лобур Т.Б.

підпис

(прізвище та ініціали)

Завідувач кафедри

Загородна Н.В.

підпис

(прізвище та ініціали)

Рецензент

підпис

(прізвище та ініціали)

м. Тернопіль – 2023

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)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Загородна Н.В.
(підпис) (прізвище та ініціали)

«__» _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Галанському Іллі Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка і налаштування системи сигналізації з дистанційним керуванням

Керівник роботи Лечаченко Тарас Анатолійович Ph.D, асистент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «03» 04 2023 року № 4/7-349

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Вимоги до програмного забезпечення

4. Зміст роботи (перелік питань, які потрібно розробити)

Загальна частина

Обґрунтування актуальності теми кваліфікаційної роботи

Аналітичний огляд існуючих рішень

Розробка технічного та робочого проекту

Аналіз технічного завдання кваліфікаційної роботи

Опис і обґрунтування вибору елементної бази

Розробка і опис функціональної схеми, алгоритму роботи

Налаштування сигналізації та написання текстів програми

Розробка інструкції з методики перевірки, функціонування (контролю, випробування) електронного пристрою

Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.02 – 19.02	<i>Виконано</i>
2.	Підбір джерел про мікропроцесори	20.02 – 27.02	<i>Виконано</i>
3.	Опрацювання джерел в галузі дослідження	28.02 – 16.03	<i>Виконано</i>
4.	Розроблення програмного коду	17.03 – 20.03	<i>Виконано</i>
5.	Тестування роботи програми та верифікація результатів	20.03 – 05.04	<i>Виконано</i>
6.	Оформлення розділу «Загальна частина»	06.03 – 17.04	<i>Виконано</i>
7.	Оформлення розділу «Розробка технічного та робочого проекту»	18.04 – 29.04	<i>Виконано</i>
8.	Оформлення розділу «Налаштування сигналізації»	30.04 – 13.05	<i>Виконано</i>
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	14.05 – 21.05	<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	<i>Виконано</i>
11.	Нормоконтроль	06.06 – 12.06	<i>Виконано</i>
12.	Перевірка на плагіат	10.06 – 15.06	<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	22.06.2023	

Студент

(підпис)*Галанський І.О.*_____
(прізвище та ініціали)

Керівник роботи

(підпис)*Лечаченко Т.А.*_____
(прізвище та ініціали)

АНОТАЦІЯ

Розробка і налаштування сигналізації з дистанційним керуванням // Кваліфікаційна робота ОР «Бакалавр» // Галанський Ілля Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-41 // Тернопіль, 2023 // С. – 54 , рис. – 20, табл. – 5 , додат. – 1.

Ключові слова: СИГНАЛІЗАЦІЯ, ДИСТАНЦІЙНЕ КЕРУВАННЯ, АРДУІНО, GSM МОДУЛЬ, НАЛАШТУВАННЯ.

Ця кваліфікаційна робота присвячена розробці технічного та робочого проекту, а також налаштуванню системи сигналізації. У загальній частині роботи обґрунтовується актуальність теми та проводиться аналіз існуючих рішень.

У розділі "Розробка технічного та робочого проекту" проводиться аналіз технічного завдання кваліфікаційної роботи та обґрунтовується вибір елементної бази. Також надається розробка функціональної та структурної схем і алгоритму роботи системи.

У розділі "Налаштування сигналізації" описується процес написання текстів програми для електронного пристрою, розробка інструкції з експлуатації пристрою та методики перевірки його функціонування.

Також в роботі приділяється увага безпеці життєдіяльності, основам охорони праці. Розділ "Долікарська допомога при пораненнях" надає відомості про надання першої допомоги при пораненнях, а "Порядок надання домедичної допомоги постраждалим при раптовій зупинці серця" розкриває процедуру надання домедичної допомоги в разі раптової зупинки серця.

Ця кваліфікаційна робота розглядає ключові аспекти розробки системи сигналізації.

ANNOTATION

Development and configuration of remotely controlled alarm systems // Thesis of educational level "Bachelor" // Halanskyi Illia // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБс-41 group // Ternopil, 2023 // P. – 54, pict. – 20, tab. – 5, add. – 1.

Keywords: ALARM SYSTEM, REMOTE CONTROL, ARDUINO, GSM MODULE, CONFIGURATION.

This qualification work is dedicated to the development of a technical and working project, as well as the configuration of an alarm system. The general part of the work substantiates the relevance of the topic and conducts an analysis of existing solutions.

In the section "Development of the Technical and Working Project," an analysis of the technical assignment of the diploma project is carried out, and the choice of the component base is justified. It also includes the development of functional and structural diagrams and the system's operation algorithm.

The section "Configuration of the Alarm System" describes the process of writing program texts for the electronic device, the development of device operation instructions, and methods for verifying its functioning.

The work also pays attention to life safety and the basics of labor protection. The section "First Aid for Injuries" provides information on providing first aid for injuries, while the section "Procedure for Providing Pre-medical Assistance to Victims of Sudden Cardiac Arrest" explains the procedure for providing pre-medical assistance in case of sudden cardiac arrest.

This qualification work considers key aspects of developing an alarm system.

ЗМІСТ

ВСТУП	7
1 ЗАГАЛЬНА ЧАСТИНА.....	9
1.1 Обґрунтування актуальності теми кваліфікаційної роботи.....	9
1.2 Аналітичний огляд існуючих рішень.....	10
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ	15
2.1 Аналіз технічного завдання кваліфікаційної роботи	15
2.2 Опис і обґрунтування вибору елементної бази.....	17
2.3 Розробка і опис функціональної схеми.....	31
2.4 Розробка алгоритму роботи системи	33
3 НАЛАШТУВАННЯ СИГНАЛІЗАЦІЇ	36
3.1 Написання текстів програми.....	36
3.2 Розробка інструкції з експлуатації електронного пристрою	44
3.3 Розробка методики перевірки, функціонування (контролю, випробування) електронного пристрою	45
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	47
4.1 Долікарська допомога при пораненнях	47
4.2 Порядок надання домедичної допомоги постраждалим при раптовій зупинці серця	49
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК А.....	55

ВСТУП

Сучасні технології стежать за швидким розвитком і внесенням змін у всі сфери нашого життя. Однією з найбільш важливих областей, що отримала значні покращення завдяки цьому прогресу, є безпека житла. Зростання інтересу до систем автоматизації та високотехнологічних пристроїв привело до появи концепції розумного дому. Розумний дім - це не просто місце для проживання, а справжній житловий простір, організований з використанням автоматизації та інтегрованих систем, що забезпечують безпеку, комфорт та ефективне використання ресурсів для мешканців.

Одним із важливих аспектів розумного дому є система сигналізації, яка забезпечує безпеку та захист мешканців і їхнього майна. У цій кваліфікаційній роботі пропонується розробка системи сигналізації з дистанційним керуванням, яка поєднує в собі передові технології та принципи розумного дому.

Метою даної кваліфікаційної роботи є розробка системи сигналізації, яка не лише забезпечує ефективне виявлення небезпеки, але й надає можливість дистанційного керування та моніторингу за станом системи. Система буде базуватися на використанні сучасних сенсорів, комунікаційних технологій та інтеграції з розумними домашніми системами. Це дозволить користувачам миттєво отримувати сповіщення про події, що відбуваються в їхньому житлі, і дистанційно керувати режимами роботи системи сигналізації.

У процесі розробки системи буде звертатися особлива увага на безпеку, надійність та простоту використання. Планується провести дослідження та експерименти з метою визначення оптимальних параметрів системи сигналізації, її функціональних можливостей та способів інтеграції з існуючими розумними системами.

Очікується, що результати даної роботи сприятимуть покращенню рівня безпеки житла та забезпечать зручні та ефективні можливості

керування системою сигналізації з використанням сучасних технологій дистанційного керування.

Зазначена робота складається з наступних розділів: аналіз сучасних систем сигналізації, розробка концепції системи сигналізації з дистанційним керуванням, проектування та реалізація системи, експериментальні дослідження та аналіз результатів, висновки та рекомендації.

Розроблена система сигналізації з дистанційним керуванням буде важливим внеском у розвиток розумних домів і сприятиме покращенню безпеки та комфорту життя мешканців.

1 ЗАГАЛЬНА ЧАСТИНА

1.1 Обґрунтування актуальності теми кваліфікаційної роботи

Метою даної кваліфікаційної роботи є розробка системи сигналізації з дистанційним керуванням на базі мікроконтролерної плати Arduino.

На сьогоднішній день багато людей стурбовані захистом свого майна, адже дотепер великий відсоток злочинів в Україні випадає саме на крадіжки.

Дивлячись на таку картину велика кількість українців хоче запобігти втрати свого майна, в наслідку чого купує охоронні системи для своїх авто, квартир, будинків тощо.

Сучасні мікропроцесорні системи використовуються для автоматизації систем управління технологічними процесами в промисловості.

Частіше за все в якості охоронної системи виступає сигналізація. Сигналізації на «ключ» вже давно усталений вид, який використовується повсюдно. Але в більшості випадків для керування і моніторингу сигналізації, це треба робити знаходячись безпосередньо біля панелі керування, що в свою чергу доставляє незручності.

В даній кваліфікаційній роботі буде вирішена проблема керування і моніторингу сигналізації з будь якого місця, маючи в руках лише телефон та зв'язок. Система сигналізації з дистанційним керуванням є рішенням даної проблеми, завдяки GSM модулю сигналізацією можна буде керувати через SMS команди, перевірити стан пристрою, увімкнути або вимкнути сигналізацію, налаштувати SMS повідомлення та дзвінки при спрацьовуванні.

Така система сигналізації з дистанційним керуванням на базі мікроконтролерної плати Arduino має ряд переваг, які роблять її привабливою для використання у різних ситуаціях. По-перше, вона дозволяє зручно керувати сигналізацією з будь-якого місця, де є зв'язок і доступ до мобільного телефону. Це означає, що власники будинків, квартир або

автомобілів зможуть контролювати стан своєї системи сигналізації та реагувати на події навіть тоді, коли вони знаходяться далеко від місця установки.

Крім того, система заснована на мікроконтролері Arduino, який є доступним, надійним та легко програмованим. Це дає можливість розробникам швидко створювати та модифікувати функціонал системи залежно від потреб користувача. Використання GSM модулю дозволяє передавати і отримувати SMS команди для керування системою, що є зручним та швидким способом взаємодії з сигналізацією.

Окрім функції керування, система також може надсилати SMS повідомлення та здійснювати дзвінки користувачеві при спрацюванні сигналізації. Це дає можливість отримати оперативну інформацію про незвичайні події і при необхідності вжити невідкладних заходів.

Застосування системи сигналізації з дистанційним керуванням може бути широким, починаючи від приватного сектору до комерційних будівель і автомобілів. Люди зможуть відчувати більшу безпеку, знаючи, що вони можуть контролювати свою систему сигналізації незалежно від свого місця перебування.

Тема цієї кваліфікаційної роботи є актуальною, адже система сигналізації з дистанційним керуванням допоможе людям контролювати та запобігати злочинам під час їх відсутності і відповідати сучасним потребам людей у безпеці та захисті.

1.2 Аналітичний огляд існуючих рішень

Серед систем сигналізації з дистанційним керуванням можна виділити наступні:

AJAX

AJAX StarterKit - це сучасна система сигналізації з можливістю дистанційного керування. Набір включає в себе різноманітні компоненти для

охоронної системи Ajax: інтелектуальний центр Ajax Hub, датчик руху Ajax MotionProtect, універсальний датчик відкриття вікон та дверей Ajax DoorProtect, а також брелок Ajax Space Control для управління системою безпеки.

На рисунку 1.1 зображений набір AJAX StarterKit.



Рисунок 1.1 – Набір AJAX StarterKit

Система може бути керована з віддалено за допомогою мобільних додатків для iOS та Android. За кілька натискань можна налаштувати датчики, активувати режим часткової охорони та перевірити стан кожного пристрою. Ajax легко підключається до пульта охорони. Запит на обслуговування можна надіслати безпосередньо з мобільного додатка – в списку вже більше 35 приватних охоронних компаній з усієї України [1].

У комплект системи Ajax StarterKit входять наступні складові:

- розумна централь Ajax Hub;
- безпроводний датчик відкриття вікон та дверей Ajax DoorProtect strong;
- безпроводний датчик руху Ajax MotionProtect

- брелок для керування охоронною системою Ajax SpaceControl;
- дріт живлення;
- ethernet дріт;
- монтажний комплект;
- керівництво по установці і експлуатації.

Технічні характеристики системи сигналізації AJAX StarterKit відображені в таблиці 1.1

Таблиця 1.1 - Технічні характеристики системи сигналізації AJAX StarterKit

Канали зв'язку	Ethernet та GSM як резервний канал
Кількість пристроїв	До 100 пристроїв на хаб
Відстань	2000 метрів.
Час роботи на резервному живленні	7 годин
Тип підключення	Безпроводне

Smart 103 (PG-103)

Безпроводна GSM + Wi-Fi сигналізація Smart 103 є сучасним, надійним і головне дешевим варіантом охорони приміщення. Ефективну роботу системи забезпечує потужний та продуктивний процесор, а також надійні Wi-Fi і GSM модулі. Весь набір бездротової сигналізації складається з нових компонентів, які відзначаються своєю практичністю. Передові моделі датчиків руху і відкриття включають в себе найкращий дизайн і високоякісні компоненти. Наявність ЖК-екрану і сенсорних клавiш робить налаштування бездротової сигналізації простим та повсякденне використання зручним.

На рисунку 1.2 зображений набір безпроводної сигналізації Smart 103 (PG-103).



Рисунок 1.2 – Набір безпроводної сигналізації Smart 103 (PG-103)

Крім традиційних функцій, які включають видзвонювання та відправлення SMS повідомлень, система має розширені можливості роботи через інтернет. При підключенні до Wi-Fi, система може повністю функціонувати не лише через мережу GSM, але й використовувати інтернет-з'єднання. Повідомлення надсилаються на смартфон з встановленим мобільним додатком. Керування системою та її налаштування також можуть бути здійснені через інтернет у мобільному додатку.

Сигналізація має широкий спектр функціональних можливостей, включаючи можливість постановки або зняття з охорони за допомогою SMS, підтримку до 100 датчиків, сповіщення про втрату живлення, можливість підключення бездротової сирени, можливість програмування постановки або зняття з охорони за розкладом, а також перегляд журналу подій на екрані системи або через мобільний додаток [2].

У комплект системи Smart 103 (PG-103) входять наступні компоненти:

- безпроводний датчик відкриття;
- блок живлення 2020В/5,1В 2,1А;
- безпроводний датчик руху;

- провідна сирена;
- контрольна панель;
- ключ – 2шт.;
- пульт дистанційного керування – 2шт.;
- інструкція на українській мові.

Технічні характеристики системи сигналізації Smart 104 (PG-104) відображені в таблиці 1.2

Таблиця 1.2 - Технічні характеристики системи сигналізації

Частота безпроводних датчиків	433 МГц 2262/1.5-4.7 М
GSM мережа	850/900/1800/1900 МГц
Вхідна напруга	12 В, 2 А
Струм в пасивному режимі	120 мА
Струм в активному режимі	500 мА
Резервний акумулятор	3,7 В/500 мАч
Сповіщення про тривогу	SMS/Дзвінок (до 5 номерів)
Кількість провідних/безпроводних зон	0/100
Макс. к-сть безпроводних датчиків	100
Макс. к-сть підключених пультів керування	100
Керування	Пульт/SMS/Дзвінок/ПО
Можливість підключення провідних зон	Немає
SMS постановка або зняття з охорони	Так
Можливість розкомплектувати	Немає
Наявність релейних виходів	Так
Протокол передачі даних	ADEMCO, ContactID
Відстань роботи датчиків	До 100 метрів
Вага	890 г

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Аналіз технічного завдання кваліфікаційної роботи

Метою даної кваліфікаційної роботи є розробка системи сигналізації з дистанційним керуванням на базі апаратної платформи Arduino.

Дана система призначена для віддаленого керування сигналізацією в період відсутності.

Структурна схема є першим абстрактним зображенням електронного пристрою, яке узагальнено відображає його принцип роботи. Структурна схема має перевагу в тому, що вона надає можливість швидко отримати уявлення про склад, структуру та виконувані функції пристрою [3].

Структурна схема пристрою зображена на рисунку 2.2

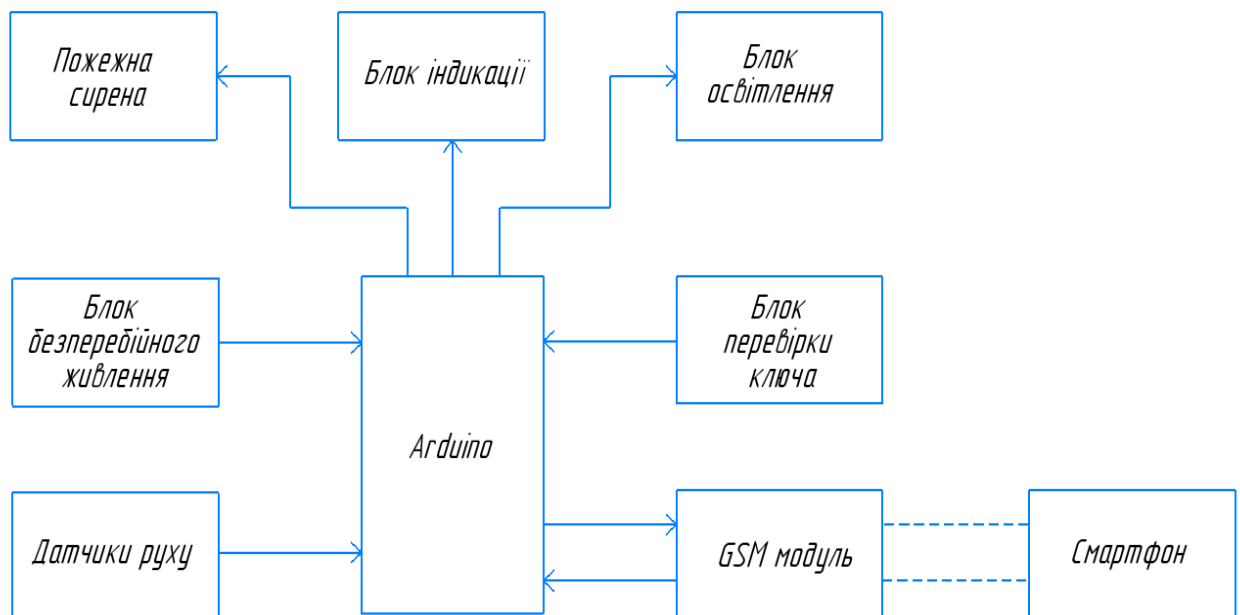


Рисунок 2.2 – Структурна схема пристрою

Система сигналізації з дистанційним керуванням містить:

- arduino(UNO) – це ядро всього приладу, призначене для зчитування, обробки і надсилання даних;

- датчик руху – призначений для виявлення руху в полі зору датчика;
- блок безперебійного живлення – складається з двох модулів, mini-usb контролера і акумулятора на 3.7 В.
- сирена – призначена для гучного звукового сповіщення.
- блок індикації – два світлодіоди що вказують на режим сигналізації.
- блок освітлення – призначений для індикації роботи реле.
- блок перевірки ключа – призначений для постановки сигналізації на охорону за допомогою домофонного ключа.
- GSM модуль – призначений для надсилання та отримування даних з смартфона.

Порядок роботи системи при виявленні руху:

- перевірка ключа.
- якщо ключ вірний – через 10 секунд ставить сигналізацію на охорону.
- якщо ключ не вірний – блокує доступ на 10 секунд, через 10 секунд дозволяє зчитати ключ знову;
- зчитування інформації з датчиків.
- якщо є рух – вмикається сирена, приходить SMS-повідомлення та три послідовних дзвінки з проміжком 20 секунд.
- вимкнути сигналізацію ключем.
- повернутись на початок.

2.2 Опис і обґрунтування вибору елементної бази

Система розроблена на основі плати Arduino Uno.

В якості мікропроцесора в системі управління використовується 8-ми розрядний мікроконтролер ATMEL – ATmega328P.

Arduino Uno базується на мікроконтролері ATmega328P. Ця платформа має 14 цифрових входів/виходів, з яких 6 можуть використовуватись як аналогові, а також 6 як виходи ШІМ. Вона оснащена кварцовим генератором з тактовою частотою 16 МГц, роз'ємом USB, силовим роз'ємом, роз'ємом ICSP для програмування та кнопкою перезавантаження.

Під час роботи з Arduino Uno необхідно підключити її до комп'ютера за допомогою USB-кабелю або ж подати живлення від адаптера AC/DC або батареї.

Нововведенням в Arduino Uno є використання мікроконтролера ATmega8U2 для забезпечення зв'язку через USB, що відрізняється від попередніх версій, де використовувався мікроконтролер FTD USB. Програмне забезпечення для розробки Arduino - це інтегроване середовище, розроблене на основі мови програмування Java, яке має модуль передачі прошивки на плату Arduino, редактор та компілятор коду [4].

Виводи Arduino Uno: Vcc, GND.

Цифрові входи/виходи: 1,2,4-16.

Аналогові входи: 19-24.

Виводи інтерфейсу I2C: 27,28.

Виводи передачі даних: 1,2,D-,D+.

Вихід еталонної аналогової напруги: AREF.

Вихід з робочою напругою входів/виходів плати Arduino Uno: IOREF.

Виводи програматора ICSP: 29-34.

На рисунку 2.3 зображено зовнішній вигляд плати Arduino Uno

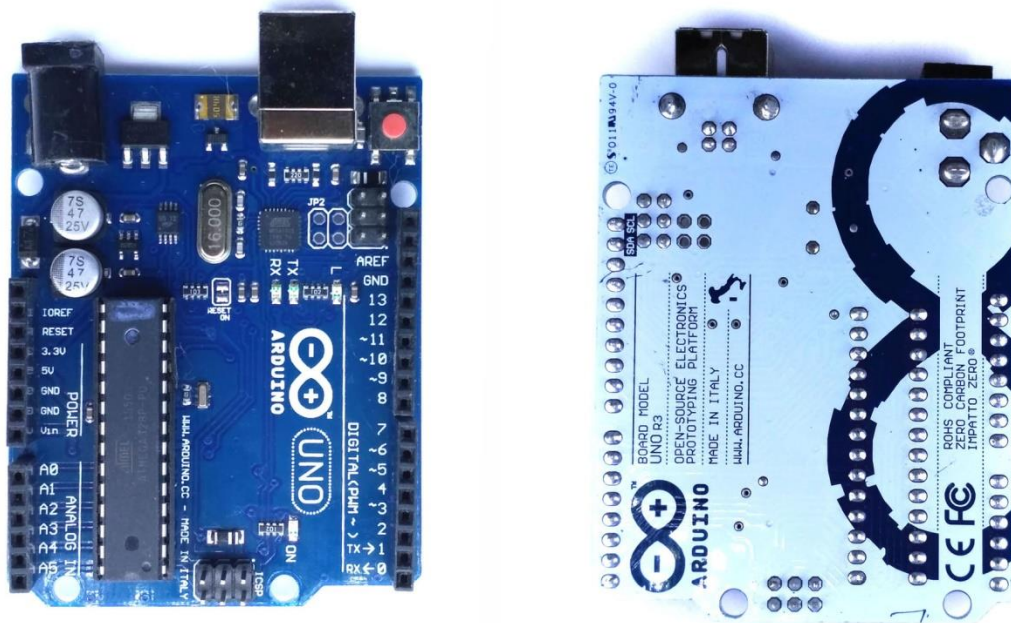


Рисунок 2.3 – Зовнішній вигляд плати Arduino Uno

Характеристики плати Arduino Uno вказані у таблиці 2.1

Таблиця 2.1 – Характеристики Arduino Uno

Мікроконтролер	ATmega328
Робоча напруга	5 В
Вхідна напруга (рекомендована)	7-12 В
Вхідна напруга (гранична)	6-20 В
Цифрові Входи/Виходи	14 (6 з яких можуть використовуватися як виходи ШІМ)
Аналогові входи	6
Постійний струм через вхід/вихід	40 мА
Постійний струм для виведення 3.3В	50 мА
Флеш – пам'ять	32 Кб (ATmega328P) з яких 0.5 Кб використовуються для завантажувача
ОЗУ	2 Кб (ATmega328P)
EEPROM	1 Кб (ATmega328P)
Тактова частота	16 МГц

Arduino Uno отримує живлення від USB або зовнішнього джерела. Вибір джерела живлення автоматичний. Якщо використовується зовнішнє живлення, воно може подаватися через блок живлення (AC/DC-перетворювач) або акумуляторну батарею. Перетворювач напруги підключається до роз'єму з діаметром 2.1 мм. Проводи від батареї можна підключити до виводів Gnd і Vin роз'єму живлення.

Плата працює при напрузі живлення від 6 В до 20 В. З при напрузі менше 7 В, вивід 5V може мати значення менше 5 В, що може призвести до нестабільної роботи платформи. При використанні напруги вище 12 В, регулятор напруги може нагріватися і пошкодити плату.

Кожен з чотирнадцяти цифрових виводів платформи Uno може бути налаштований як вхід або вихід за допомогою функцій pinMode(), digitalWrite() і digitalRead(). Ці виводи працюють при напрузі 5 В. Кожен вивід має вбудований навантажувальний резистор (за замовчуванням вимкнений), який становить 20-50 кОм і може пропускати струм до 40 мА [5].

На рисунку 2.4 зображено умовне графічне позначення Arduino Uno.

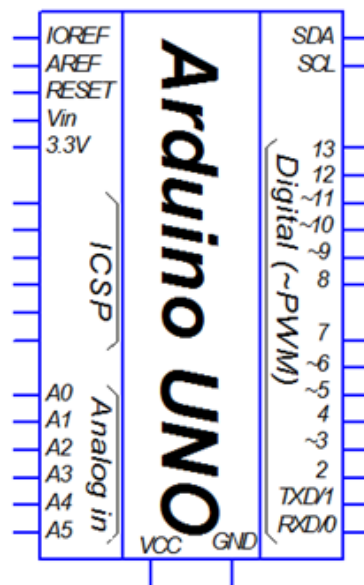


Рисунок 2.4 – Умовне графічне позначення Arduino Uno

Маючи на увазі платформу Uno, доступні шість аналогових входів з роздільною здатністю 10 біт, що означає, що вони можуть приймати 1024 різних значень. Зазвичай ці виводи працюють у діапазоні від 0 до 5 В відносно землі. Однак, є можливість змінити верхню межу вимірювання, використовуючи вивід AREF і функцію `analogReference()`.

Arduino Uno вбудований із самовідновлювальним запобіжником (автоматом), який захищає порт USB комп'ютера від струмів короткого замикання та надструмів. Хоча більшість комп'ютерів мають власний захист, цей запобіжник надає додатковий рівень безпеки. Він спрацьовує, якщо струм перевищує 500 мА через USB-порт, тим самим розриваючи ланцюг, доки струм не повернеться до нормальних значень.

Плата Uno має розміри 6,9 см у довжину та 5,3 см у ширину. Роз'єми USB і силовий роз'єм виступають за межі цих розмірів. Плата має чотири отвори, які дозволяють її закріпити на поверхні. Відстань між цифровими виводами 7 і 8 становить 0,4 см, а між іншими виводами ця відстань складає 0,25 см.

Технічний опис контролера ATmega328P:

- висока продуктивність з низькою потужністю;
- прогресивна RISC архітектура;
- 131 Потужна інструкція;
- 32 x 8 загального призначення робочих регістрів;
- повністю статичні операції;
- до 20 MIPS пропускна здатність у 20 МГц;
- високо витривалі енергозалежності сегменти пам'яті;
- 4/8/16/32К байтова система само-програмованої флеш-пам'яті (ATmega48P/88P/168P/328P);
- 256/512/512/1К БАЙТ EEPROM (ATmega48P/88P/169P/328P);
- 512/1К/1/2К байт внутрішня SRAM (ATmega48P/88P/168P/328P);
- цикл запису/стирання: 10 000 Flash / 100 000 EEPROM;

- зберігання даних: 20 років при високій температурі / 100 років при низькій температурі;
- необов'язковий розділ завантажувального коду з незалежними бітами блокування;
- блокування програмування використовується для безпеки програмного забезпечення;
- периферійні особливості;
- два 8-розрядних таймера/лічильники з роздільним прескалером і режимом порівняння;
- один 16-розрядний таймер/лічильник з окремим попереднім дільником, режимом порівняння;

Режими:

- 6-канальний ШПМ;
- 8-канальний 10-бітний АЦП в TQFP і QFN / МФ пакет;
- 6-канальний 10-бітний АЦП в PDIP пакеті;
- програмований послідовний USART;
- master/slave SPI послідовний інтерфейс;
- програмований сторожовий таймер з окремим вбудованим генератором;
- вбудований аналоговий компаратор;

Спеціальні особливості мікроконтролера:

- скидання по включенню живлення;
- внутрішній калібрований генератор;
- зовнішні і внутрішні джерела переривань;
- режим сну: скорочення АЦП, енергозбереження, при відключенні живлення, в режимі очікування;

I / O і пакети:

- 23 програмованих входів/виходів;
- Робоча напруга:
- від 40 °С до 85 °С;

- активний режим: 0.3 мА;
- при відключенні живлення режим: 0,1 мкА;
- режим економії енергії: 0,8 мкА (у тому числі 32 кГц RTC);
-

На рисунку 2.5 зображено виводи контролера АТmega328Р.

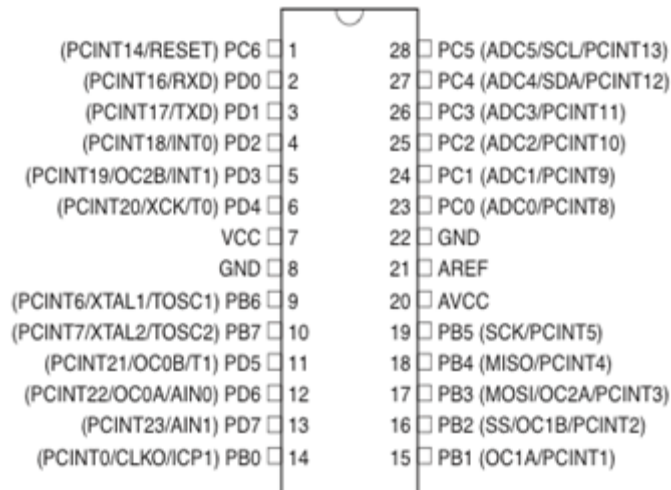


Рисунок 2.5 – Виводи контролера АТmega328Р

Датчик руху

Для виявлення переміщення на території використовується PIR – датчик руху. На рисунку 2.6 зображений зовнішній вигляд PIR - Інфрачервоного датчика руху для Arduino. Датчик дозволяє виявляти рух на відстані до 7 метрів і більше [6].



Рисунок 2.6 – Зовнішній вигляд PIR датчика руху



Рисунок 2.7 – Умовне позначення датчика руху

Основні параметри датчика зображені у таблиці 2.2

Таблиця 2.2 – Основні параметри датчика руху

Розміри	Приблизно 3.2см x 2.4 см x 1.8см
Напруга живлення DC	4.5V – 20V
Струм на OUT	<60µA
Напруга на виході	Високі і низькі рівні в 3.3V TTL логікою
Дистанція виявлення	3 – 7м (настроюється)
Кут виявлення	До 120° - 140° (залежно від конкретного датчика і лінзи)
Тривалість імпульсу при виявленні	5 – 200сек. (настроюється)
Час блокування до наступного заміру	2.5сек. (але можна змінити заміною SMD-резисторів)
Робоча температура	-20 - +80 °C
Режим роботи	L – одиночний захоплення, H - повторювані вимірювання

Підключити датчик до Arduino досить просто (див. рис 2.8). Крім живлення, сигнальний вивід підключається до будь-якого цифрового виводу Arduino, і потребує 10 кОм підтягує резистори між сигналом і 5В.

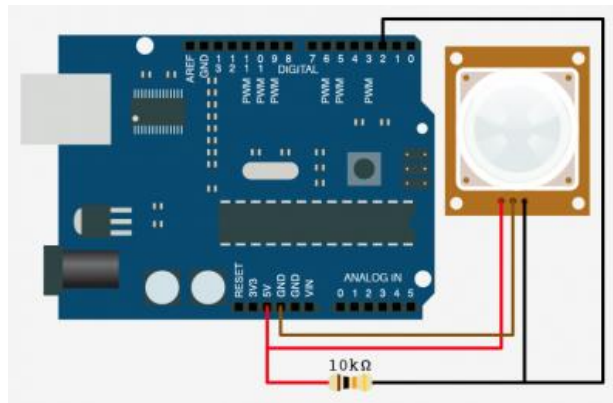


Рисунок 2.8 – Підключення датчика руху до Arduino

PIR датчик руху має три виводи:

- GND: "-" живлення;
- VCC: "+" живлення;
- OUT: вивід вихідного сигналу.

GND на будь-який з GND пінів Arduino.

VCC на +5 вольт на Arduino.

OUT на будь-який з цифрових входів/виходів Arduino (приєднане до 2).

GSM модуль

GSM модуль SIM800L стільникового зв'язку, використовується для отримання і передачі даних за допомогою SMS повідомлень. На рисунку 2.9 зображений зовнішній вигляд GSM модулю SIM800L [7].



Рисунок 2.9 – Зовнішній вигляд GSM модулю SIM800L



Рисунок 2.10 – Умовне позначення GSM модулю SIM800L

Основні параметри модуля зображені у таблиці 2.3

Таблиця 2.3 – Основні параметри GSM модуля

1	2
Напруга живлення	Від 3.4В до 4.4В
Рекомендована напруга живлення	4В
Струм режиму очікування	0.7 мА
Максимальний струм	500 мА
Максимальна напруга високого рівня інтерфейсу UART	2.8 В
Швидкість UART	1200-115200 бод
Робочі діапазони	EGSM900, DCS1800, GSM850, PCS1900
Потужність передачі DCS1800, PCS1900	1 Вт
Потужність передачі GSM850, EGSM900	2 Вт
Режим мережі	2G
Опір підключеного динаміку	8 Ом
Мікрофон	Електретний

Продовження таблиці 2.3

1	2
Управління командами AT через UART	3GPP TS 27.007, 27.005 SIMCOM enhanced AT Commands
Відправка та отримання GPRS даних	TCP/IP/ HTTP тощо
Максимальна швидкість передачі GPRS даних	85.6 Кбод
Кодування GSM протокол	CS-1, CS-2, CS-3 і CS-4
Підтримка пакетної передачі широкомовного каналу керування (PBCCH) CSD на швидкостях	2.4, 4.8, 9.6 і 14.4
Підтримка годин реального часу	RTC
Підтримує SIM-карти з живленням	3В і 1.8 В
Робоча температура	30 – 70 °С
Розміри	25 x 25 мм

На рисунку 2.11 зображено підключення GSM модуля SIM800L до плати Arduino

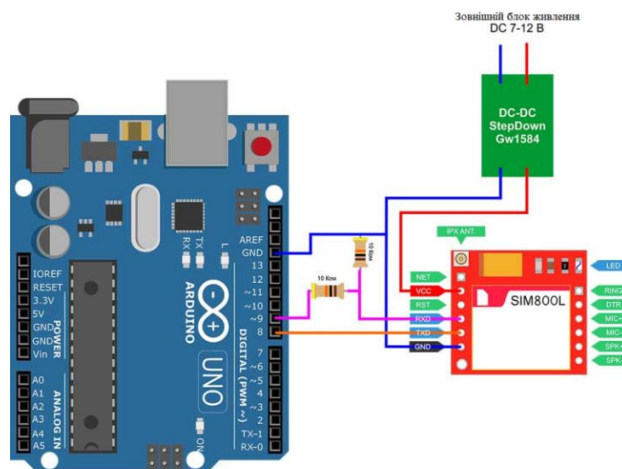


Рисунок 2.11 – Підключення GSM модулю SIM800L до Arduino

SIM800L має такі виводи:

- GND: "-" живлення;
- VCC: "+" живлення;
- DTR: додатковий сигнал UART;
- MICP: з'єднується з мікрофоном;
- RST: скидання;
- MICN: з'єднується з мікрофоном;
- RXD: до контакту TX мікроконтролера;
- SPKP: з'єднується з динаміком;
- TXD: до контакту RX мікроконтролера;
- SPKN: з'єднується з динаміком.

Блок реле

В системі використовується модуль реле, на основі реле SRD(T73).

Зовнішній вигляд модуля реле зображений на рисунку 2.12.

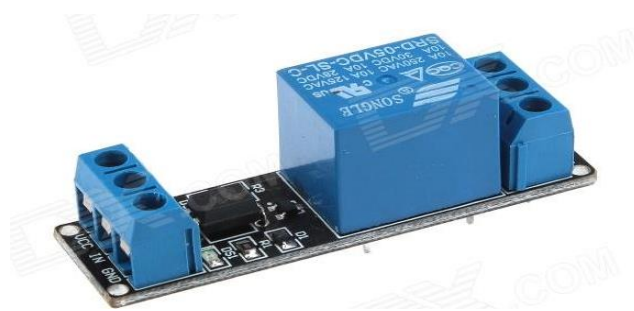


Рисунок 2.12 – Зовнішній вигляд модуля реле

Електромагнітне реле - це електричний перемикач, що працює на основі впливу електромагніту. При проходженні електричного струму один або декілька електричних контактів, відокремлених від котушки, переключаються, змінюючи стан реле. Це дозволяє виконувати комутацію електричних навантажень. Електромагнітне реле можна розглядати як універсальний перемикач для аналогових та імпульсних сигналів [8].

Характеристики SRD-05VDC-SL-C:

- матеріал контактів: сплав срібла з напиленням золота;
- робоча напруга: 5В;
- опір обмотки: 60 Ом;
- макс. Струм і напруга на перемикаючій групі контактів: 10 А 28 V DC; 10 А 124 V AC; 10 А 30V DC; 10 А 250V AC;
- зносостійкість: 100000 циклів;
- температура: -25°C + 70°C;
- габаритні розміри: 19.2x15.5x5.2 мм.
- опір контакту: 120 мОм;
- опір ізоляції: 120 мОм;

На рисунку 2.13 умовне позначення блоку реле.

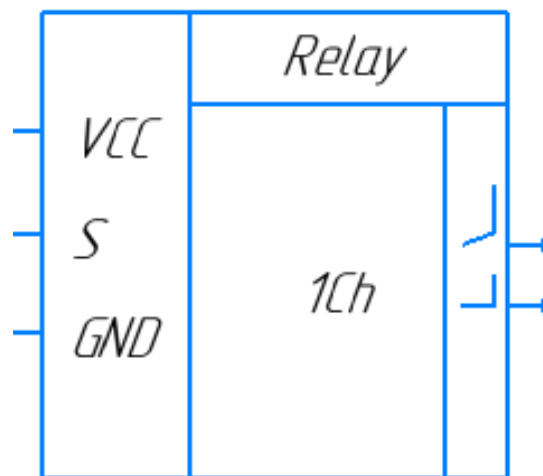


Рисунок 2.13 – Умовне позначення блоку реле

Для підключення зовнішніх пристроїв в систему через реле, користувачу потрібно приєднати зовнішнє живлення пристрою до виходу «2» і вивід живлення пристрою до виводу «3» (див. рис. 2.13).

Вивід «S» призначений для прийому керуючого сигналу.

Підключення датчика до Arduino (див. рис 2.14):

- VCC на +5 вольт на Arduino;
- IN на будь-який з цифрових входів / виходів Arduino;
- GND на будь-який з пінів GND Arduino.

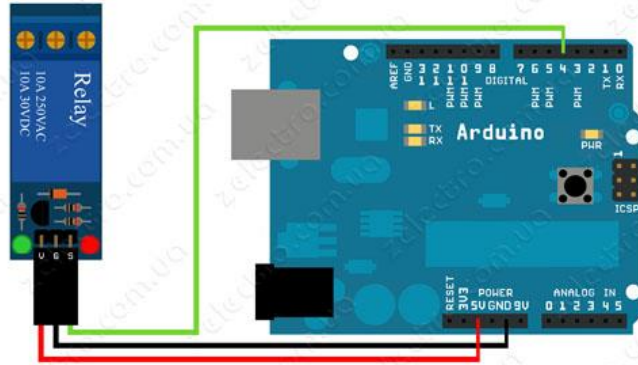


Рисунок 2.14 – Підключення блоку реле до плати Arduino

Mini USB контролер

Універсальний контролер заряду для li-ion акумуляторів (див. рис. 2.15) є компактною платою з USB портом, яка призначена для зарядки різних типів батарей та мобільних пристроїв. За допомогою шнура mini-USB здійснюється процес зарядки.

Усередині контролера розташовані світлодіоди, які випромінюють різні кольори, відображаючи рівень заряду акумулятора. При підключенні акумулятора до контролера, червоний світлодіод відразу загориться, а синій почне миготіти. Це дозволяє завжди бути в курсі рівня заряду акумулятора. Контролер зарядки літєвих акумуляторів працює в широкому діапазоні температур, від -15 до 90 °C, і має захист від короткого замикання.

Під час повної розрядки червоний індикатор вимикається, а синій загоряється. При повному заряді батареї, світлодіод стає зеленим. Контролер дозволяє регулювати струм зарядки від 80 до 850 мА за допомогою

резистора. Крім того, контролер автоматично вимикається при досягненні напруги 4 вольти [9].



Рисунок 2.15 – Mini USB контролер

Основні технічні характеристики:

- вхідний інтерфейс: Mini USB
- вхідна напруга: 4.5 - 5.5 В
- напруга заряду: 4.2 В
- струм: 1 А
- робоча температура -10°C - +85°C
- габарити: 26 x 17 x 3 мм

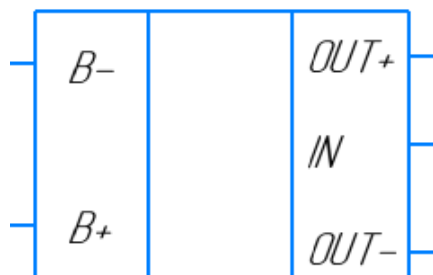


Рисунок 2.16 – Умовне позначення Mini USB контролера

Підключення контролера до Arduino (див. рис. 2.17):

- VCC +5V на Arduino;
- IN на будь-який з цифрових входів / виходів Arduino;
- GND на будь-який з пінів GND Arduino.

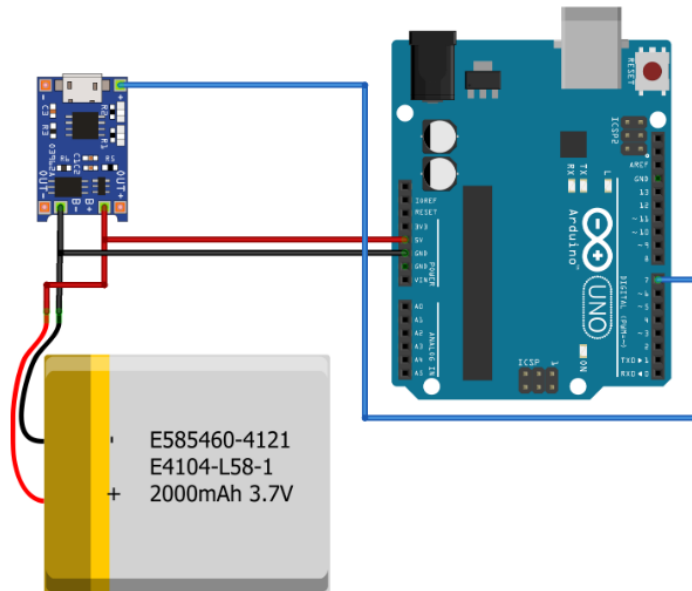


Рисунок 2.17 – Підключення контролера Mini USB до Arduino

2.3 Розробка і опис функціональної схеми

Функціональна схема є одним з ключових проектних документів, що визначає функціональну структуру системи, що розробляється, а також окремих модулів, що її складають. Її основна мета - детально відобразити структуру пристрою, його основні блоки, вузли, частини та зв'язки між ними.

Функціональна схема надає чітке уявлення про спосіб підключення та взаємодію компонентів пристрою [10].

AM1 – Mini-USB контролер TP4056;

BA1 – Пожежна сирена;

DM1,DM2,DM4,DM5,DM6 – Датчики руху HC-SR501;

DM3 – Мікропроцесор Arduino Uno;

DM7 – GSM модуль SIM800L;

HL1,HL2 – Світлодіоди індикації;

K1- Модуль реле G3MB-202P;

- R1,R2,R3,R4,R5,R6 – Резистори;
 VT1 – N-канальний транзистор;
 XS1 – Літій-іонний акумулятор 3,7V 2000mAh
 XS2 – Блок зчитування ключа;
 XS3 – Зарядне 5V;
 XS4 – Блок освітлення(лампочка).

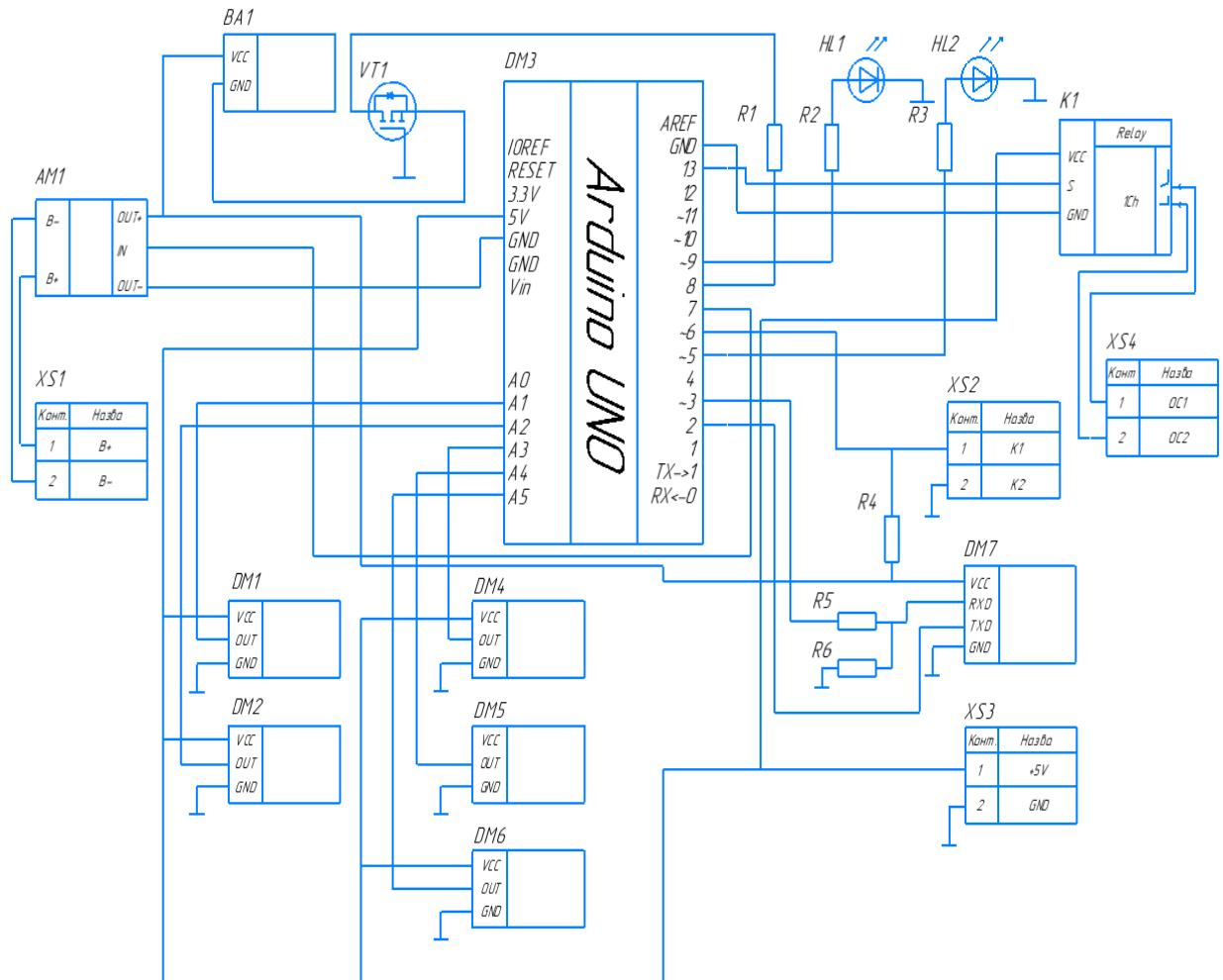


Рисунок 2.18 – Функціональна схема пристрою

2.4 Розробка алгоритму роботи системи

Для роботи з GSM модулем використовується віртуальний UART (mySerial) – бібліотека «SoftwareSerial.h». Бібліотека «EEPROM.h» використовується для запису в енергозалежну пам'ять. Для зчитування ключів використовується бібліотека «OneWire.h».

- 1) Оголошуються піни для підключення модулів.
- 2) Оголошується номер на який буде здійснюватися дзвінок якщо спрацює сигналізація.
- 3) Додається декілька номерів через кому, з котрих буде дозволено керування по СМС.
- 4) Оголошується номер для перевірки балансу SIM карти GSM сигналізації.
- 5) Оголошується номер на який будуть приходити SMS звіти.
- 6) Оголошується SMS команди для керування сигналізацією.
- 7) Додається код ключ.
- 8) Виставляється 10 секунд паузи після зчитування ключа при постановці на охорону.
- 9) Виставляється 10 секунд паузи для повторного зчитування ключа
- 10) Виставляється 20 секунд час роботи сирени, після відключення датчиків.
- 11) Виставляється 20 секунд паузу між дзвінками при спрацьовуванні сигналізації
- 12) Виставляється швидкість обміну даними з комп'ютером
- 13) Виставляється швидкість обміну даними з модемом.
- 14) Оголошується значення заряду батареї в відсотках, при якому відправиться SMS про розряджену батарею.
- 15) Виставляється паузу 3 хвилини між серіями дзвінків, при постійному спрацьовуванні кінцевика.
- 16) Оголошуються змінні для роботи з бібліотекою EEPROM.

- 17) Оголошуються змінні для зберігання точок відліку.
- 18) Оголошується змінна для зберігання відповіді модуля.
- 19) Оголошується час останнього оновлення.
- 20) Виставляється перевірка кожну хвилину.
- 21) Додається функція очікування відповіді отриманого результату.
- 22) Оголошується змінна для зберігання результату.
- 23) Оголошується змінна для відстеження тайм ауту.
- 24) Очікування відповіді 10 секунд, якщо прийшла відповідь або настав тайм аут, то `resp = SIM800.readString();`
- 25) Якщо прийшов тайм аут, то `Serial.println("Timeout...");`
- 26) Якщо є, що зчитувати `_resp = SIM800.readString();`
- 27) Оголошується змінна для зберігання результату.
- 28) Дублюється команда в монітор порту.
- 29) Дублюється відповідь в монітор порту.
- 30) Відправляється SMS.
- 31) Якщо телефон в білому списку то використовуються SMS команди.
- 32) Оголошується набір функцій для запису налаштувань в енергонезалежну пам'ять при першому запуску.
- 33) Видаляються всі SMS, для економії пам'яті.
- 34) Вмикається текстовий режим SMS й одразу зберігається значення.
- 35) Вмикається прийом спеціальних повідомлень й одразу зберігається значення.
- 36) Скидається таймер.
- 37) Налагоджується керування по UART.
- 38) Оголошуються команди керування по UART.
- 39) Перевірка наявності нових повідомлень.
- 40) Видаляється запит зчитування непрочитаних повідомлень
- 41) Скидається таймер.
- 42) Видаляються всі прочитані повідомлення.
- 43) Отримується відповідь від модему для аналізу.

Загальний алгоритм роботи зображено на рисунку 2.19

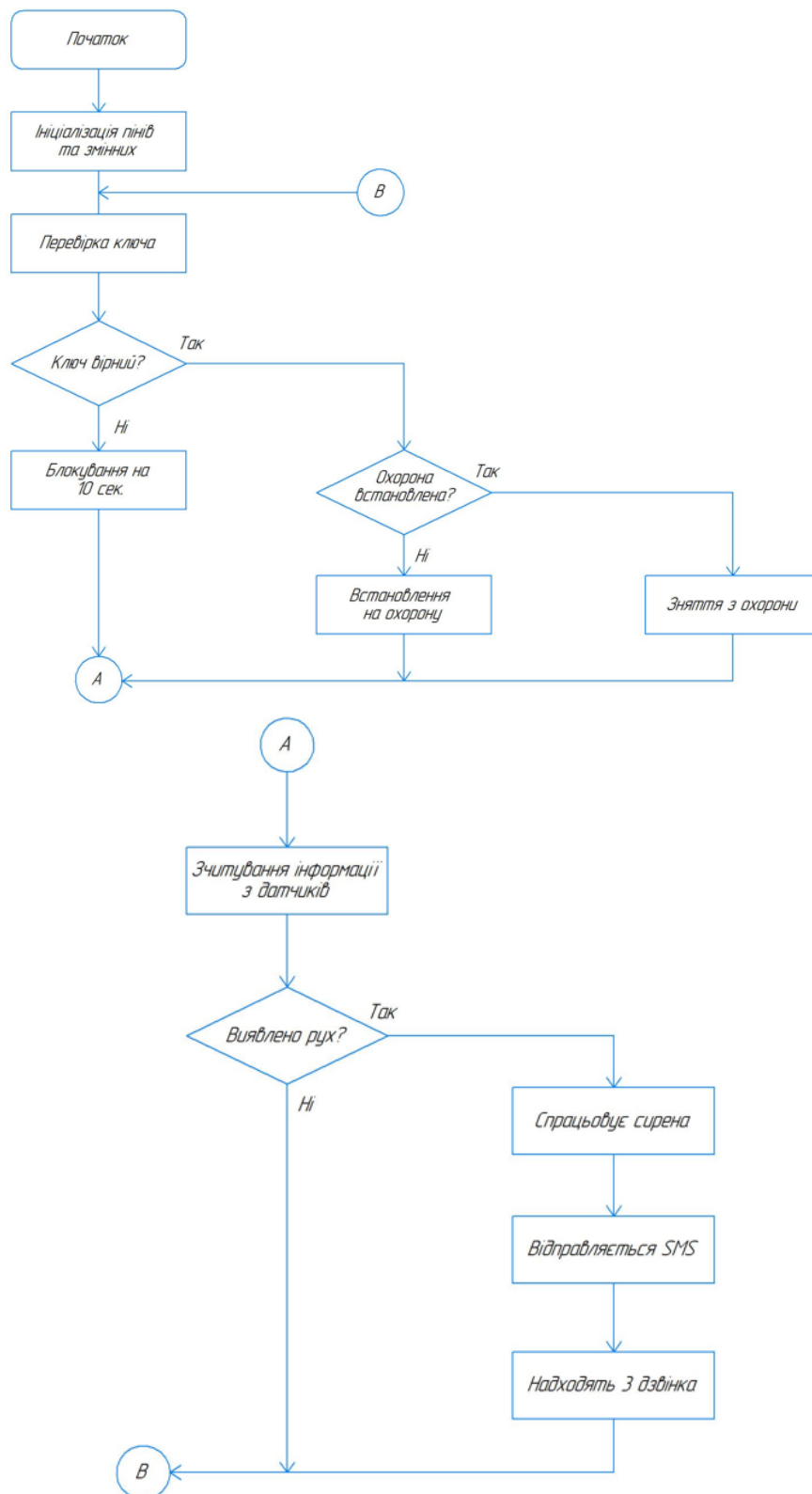


Рисунок 2.19 – Загальний алгоритм роботи

3 НАЛАШТУВАННЯ СИГНАЛІЗАЦІЇ

3.1 Написання текстів програми

На початку підключаються бібліотеки для роботи з пристроєм, для цього використовуються команди [11]:

```
#include<OneWire.h>    //бібліотека для зчитування ключів
#include "EEPROM.h"     //бібліотека для запису в енергозалежну
пам'ять
```

```
#include<SoftwareSerial.h> //бібліотека для створення другого
програмного UART для зв'язку с GSM модулем;
```

Оголошуються підключення пінів за допомогою наступних команд:

```
const int key_led = 5;    //Пін 5 для підключення світлодіода для
індикації про постанову на охорону
```

```
const int key_pin = 6;    //Пін 6 для зчитування ключа таблетки
RW1990, TM1990A
```

```
OneWire key(key_pin);
```

```
const int rx_prog = 2;    //програмний rx для підключення GSM модуля
SIM800L
```

```
const int tx_prog = 3;    //програмний tx для підключення GSM модуля
SIM800L
```

```
SoftwareSerial SIM800(rx_prog, tx_prog); //програмні RX, TX для
зв'язку з модулем SIM800L
```

```
const int led_2 = 9;    //світлодіод для індикації сигналізації
```

```
const int a1 = 15;    //лінія A1 для підключення датчика
```

```
const int a2 = 16;    //лінія A2 для підключення датчика
```

```
const int a3 = 17;    //лінія A3 для підключення датчика
```

```
const int a4 = 18;    //лінія A4 для підключення датчика
```

```
const int a5 = 19;    //лінія A5 для підключення датчика
```

```
const int speaker = 8;    //контакт для підключення сирени
```

```

const int set_220 = 7; //контакт для визначення наявності живлення
в мережі
const int relay = 13; //контакт для підключення тведротілого реле
або релейного модуля;
Оголошуються номери з яких буде дозволено здійснювати керування:
String phon = "+38097xxxxxxx"; //1 номер на який буде здійснюватися
дзвінок якщо спрацює сигналізація
String phones = "+38096xxxxxxx, +38067xxxxxxx, +48610xxxxxxx"; //1
або декілька номерів через кому, з котрих буде дозволено керування по
СМС
String balance = "*111#"; //1 номер для перевірки балансу Sim карти
GSM сигналізації
String SMS_phone = "+38097xxxxxxx"; //1 номер на який будуть
приходити SMS звіти;
Оголошуються команди за допомогою яких буде здійснюватися
керування сигналізацією:
String sms_otchet = "Ktr3"; //запит SMS про стан сигналізації,
датчиків сигналізації реле
String sig_on = "Sig1"; //поставити сигналізацію на охорону
String sig_off = "Sig0"; //зняти сигналізацію з охорони
String sms_on = "Sms1"; //ввімкнути відправку SMS при
спрацюванні сигналізації
String sms_off = "Sms0"; //вимкнути відправку SMS при
спрацюванні сигналізації
String ring_on = "Ring1"; //ввімкнути вихідний виклик при
спрацюванні сигналізації
String ring_off = "Ring0"; //вимкнути вихідний виклик при
спрацюванні сигналізації
String ballanse = "Bal"; //перевірка балансу SIM карти
String relay_on = "Rel1"; //команда для включення реле

```

```

String relay_off = "Rel0"; //команда для виключення реле;
Оголошуємо код ключа(або декількох ключів) яким буде дозволено
ставити сигналізацію на охорону:
String open_key = "1f34bf6000f7"; //1 дозволений код ключ (для
додавання декількох ключів коди записуються через кому);
Оголошується час дій та команд системи:
int sig_pause = 10; //10 секунд пауза після зчитування ключа при
постановці на охорону
int block_pause = 10; //10 секунд пауза для повторного зчитування
ключа, після того як був зчитаний невірний ключ
long spiker_time = 20; //20 секунд час роботи сирени, після
відключення датчиків
longring_time = 20; //20 секунд пауза між дзвінками при
спрацьовуванні сигналізації
int batery_limit = 50; //значення заряду батареї в відсотках, при якому
відправиться СМС про розряджену батарею (оптимально від 30% до 60%)
int ring_x = 3; //пауза 3 хвилини між серіями дзвінків, при
постійному спрацьовуванні кінцевика (від 1 до 32000 хвилин);
Оголошуються змінні для роботи з EEPROM:
int flag_EEPROM, start_FLAG = 2, address_FLAG = 0, address_SMS =
2, address_SIG = 4, address_RING = 6, address_REL = 8;
Оголошуються змінні для зберігання точок відліку, для таймерів:
unsigned long timing10, timing11, timing12, timing13, timing14, timing15,
timing16, timing17;
Оголошуються SMS команди:
if(msgbody == sms_otchet){ Serial.println(sms_otchet); delay(300); k=3;}
//запит SMS про стан сигналізації, і датчиків сигналізації
if(msgbody == sms_off){ Serial.println(sms_off); delay(300); sms_st = 0;
EEPROM.put(address_SMS, sms_st);} //вимкнути відправку SMS
при спрацьовуванні сигналізації

```



```

    if(msgbody == sms_on){ Serial.println(sms_on); delay(300); sms_st = 1;
    EEPROM.put(address_SMS, sms_st);} //ввімкнути відправку SMS
при спрацьовуванні сигналізації
    if(msgbody == ring_off){Serial.println(ring_off); delay(300); ring_st = 0;
    EEPROM.put(address_RING, ring_st);} //вимкнути вихідний виклик
при спрацьовуванні сигналізації

    if(msgbody == ring_on){Serial.println(ring_on); delay(300);ring_st =
1;ring = 0;
    EEPROM.put(address_RING, ring_st);} //ввімкнути вихідний виклик
при спрацьовуванні сигналізації
    if(msgbody == sig_off){ Serial.println(sig_off); delay(300);sig_st = 0;
    EEPROM.put(address_SIG, sig_st);} //вимкнути сигналізацію
    if(msgbody == sig_on){ Serial.println(sig_on); delay(300); sig_st = 1;
    EEPROM.put(address_SIG, sig_st);} //ввімкнути сигналізацію
    if(msgbody == relay_off){ Serial.println(relay_off); delay(300);rel_st = 0;
    EEPROM.put(address_REL, rel_st);} digitalWrite(relay, rel_st);
//вимкнути реле
    if(msgbody == relay_on){ Serial.println(relay_on); delay(300); rel_st = 1;
    EEPROM.put(address_REL, rel_st);} digitalWrite(relay, rel_st);
//ввімкнути реле
    if(msgbody == ballanse){ Serial.println(ballanse); //перевірка балансу
SIM карти
    SIM800.println("AT+CUSD=1,\"\" + balance + "\"");}
    }else {
//Serial.println("Unknown phonenumber");
    Оголошуються команду для відправки SMS та функція для
зчитування ключа:
    SIM800.print((char)26); //команда для відправки SMS
    delay(300);

```

```

void read_Button_KeY(){ //функція для зчитування ключа
if(flag_Button_KeY == 1){
if (!key.search (arr)) {
    key.reset_search(); delay(50);
return; }
for (byte x = 0; x < 8; x++) { read_key = read_key + String(arr[x], HEX);}
byte crc; crc = key.crc8(arr, 7);
Serial.println(read_key);
if (read_key.length() > 10 && open_key.indexOf(read_key) > -1) {
    timing15 = millis(); flag_Button_KeY_P = 1; flag_Button_KeY = 0;
Serial.println("Key OK!");
    digitalWrite(key_led, HIGH);
    }else{
        timing17 = millis(); flag_Button_KeY = 0; Serial.println("Key
ERROR!");
    }
    read_key = "";
    }};

```

Оголошується набір функцій для записування в енергонезалежну пам'ять при першому запуску, та швидкість обміну даних з комп'ютером і модемом:

//набір функцій для запису налаштувань в енергонезалежну пам'ять при першому запуску

```

EEPROM.get(address_FLAG, flag_EEPROM);
if(flag_EEPROM < start_FLAG){
EEPROM.put(address_FLAG, start_FLAG);
EEPROM.put(address_SMS, sms_st);
EEPROM.put(address_SIG, sig_st);
EEPROM.put(address_RING, ring_st);
EEPROM.put(address_REL, rel_st);

```



```

    }
    EEPROM.get(address_SMS, sms_st);
    EEPROM.get(address_SIG, sig_st);
    EEPROM.get(address_RING, ring_st);
    EEPROM.get(address_REL, rel_st);
    Serial.begin(9600); // Швидкість обміну даними з
комп'ютером
    SIM800.begin(9600); // Швидкість обміну даними з
модемом
    analogReference(INTERNAL);
    Оголошуються дії індикації при спрацьовуванні датчика, за
допомогою наступних команд:
    // якщо датчик спрацює
    if (ring == 0 && led_2_ON == 200 && ring_fl == 0)
    { signalingS = "ПОПЫТКА VZLOMA"; digitalWrite(speaker, HIGH);
    if (ring_st == 1){ring ++; Serial.println("<<ring ++;>>");}
    if(sms_st == 1){send_mesag++; }
    timing10 = millis(); Serial.println("VZLOM");
    } else { signalingS = signaling_OK;
    if (millis() - timing12 > spiker_time){timing12 = millis();
digitalWrite(speaker, LOW);};
    // 3 дзвінка при спрацьовуванні датчиків
    if (ring > 0 && millis() - timing10 > ring_time * 1000 && ring_st == 1
&& ring_fl == 0){
    timing10 = millis(); ring++;
    String ring1 = String(ring-1);
    Serial.println("<< Ring " + ring1 + " >>");
    SIM800.println("ATD" + phon + ";");
    }
    if (ring > 3) {ring = 0; send_mesag = 0; ring_fl = 1;}

```

```

//опитування датчиків
if(stat_a1 == LOW && stat_a2 == LOW && stat_a3 == LOW && stat_a4
== LOW && stat_a5 == LOW)
{blinker = 1;}else{blinker = 0;}
//блінк для індикації сигналізації
if(sig_st == 0 && blinker == 1){led_2_ON = 1000; led_2_OFF = 1000;}
if(sig_st == 1 && blinker == 1){led_2_ON = 3000; led_2_OFF = 100;}
Оголошується повідомлення про розряджену батарею:
if(state_Batery > 11 && state_Batery <
batery_limit){send_mesag_state_Batery++;}
if (send_mesag_state_Batery == 1){
// Serial.println("<<SMS Batery < 50%>>");
numbr_sms();
SIM800.println("Set = " + String(v) + " volt");
delay(300);
SIM800.print("Battery = " + String(state_Batery) + "%");
go_sms();
}
if (send_mesag_state_Batery > 2) {send_mesag_state_Batery = 2;}
if(state_Batery > 80){send_mesag_state_Batery = 0;};
Оголошуються команди для керування сигналізацією за допомогою
ключа таблетки:
//постановка на охорону ключем таблеткою
if(sig_st == 0 && flag_Button_KeY_P == 1 && millis() - timing15 >
sig_pause * 1000){
sig_st = 1; flag_Button_KeY_P = 0; //затримка спрацьовування після
зчитування ключа, виставляється змінною sig_pause на початку скетчу
flag_Button_KeY = 1; //дозволяємо зчитування ключа
//зняття з охорони ключем таблеткою
if(sig_st == 1 && flag_Button_KeY_P == 1){

```

```

        sig_st = 0; flag_Button_KeY_P = 0; timing16 = millis();
        EEPROM.put(address_SIG, sig_st);           //зберігаємо стан в
енергонезалежну пам'ять
        digitalWrite(key_led, LOW);
    }
    if(sig_st == 0 && flag_Button_KeY_P == 0 && flag_Button_KeY == 1
&& millis() - timing16 > 3000){
        flag_Button_KeY = 1;                       //через 3 секунди після зняття з
охорони дозволяємо зчитування ключа
    }
    //при невірному коді ключа не дозволяємо зчитувати ключ впродовж
часу block_pause
    if(flag_Button_KeY == 0 && millis() - timing17 > block_pause * 1000){
        flag_Button_KeY = 1;}                     //дозволити сканування ключа після
закінчення часу block_pause
    };

```

3.2 Розробка інструкції з експлуатації електронного пристрою

Для того, щоб система виконувала свої функції, перш за все до Arduino потрібно приєднати елементи: такі як датчик руху, GSM модуль, mini USB контролер, акумулятор. Підключення потрібно проводити відповідно до функціональної схеми. Після цього потрібно перевірити правильність з'єднання. Коли перевірка зроблена, модуль, потрібно поставити у контрольовану зону. Пристрій зчитує сигнал з датчика руху. Як тільки об'єкт з'явиться в полі його дії, включиться гучна сирена, а на телефон надійде 3 дзвінка і SMS звіт.

Для роботи приладу необхідно купити SIM карту і мати на ній позитивний баланс коштів.

Два світло діоди служать індикацією сигналізації, якщо на плату подається живлення червоний світлодіод постійно горить. Після встановлення сигналізації на охорону за допомогою ключа, білий світлодіод горить 10 секунд, після чого з періодичністю в три секунди починає блимати червоний світло діод [12].

Лампа служить індикацією роботи реле, яку можна включити за допомогою SMS команди.

- SMS команди для керування сигналізацією:
- Ktr3 – запит про стан сигналізації, датчиків і реле
- Sig1 – поставити сигналізацію на охорону
- Sig0 – зняти сигналізацію з охорони
- Sms1 – включити відправку SMS при спрацьовуванні сигналізації
- Sms0 – виключити відправку SMS при спрацьовуванні сигналізації
- Ring1 – включити вихідний дзвінок при спрацьовуванні сигналізації
- Ring0 – виключити вихідний дзвінок при спрацьовуванні сигналізації

- Bal – перевірка балансу SIM карти
- Rel1 – команда для включення реле
- Rel0 – команда для виключення реле

3.3 Розробка методики перевірки, функціонування (контролю, випробування) електронного пристрою

Для тестування і налагодження системи використовуються віртуальний термінал вбудований в середовище Arduino. Значення з датчику руху виводяться у вікні терміналу.

У автоматизованій системі сигналізації з дистанційним керуванням можливі такі види несправності:

1) Не приходять дзвінки і SMS звіти на смартфон:

- перевірити правильність підключення датчика до Arduino;
- перевірити чи є кошти на сім карті;
- перевірити чи справний смартфон;
- перевірити чи подається живлення на плату.

2) Помилкові спрацьовування:

- перевірити датчик руху на наявність пошкоджень;
- зменшити чутливість датчика.

3) Не функціонує сирена:

- перевірити правильність підключення датчика до Arduino;
- перевірити сирену на наявність пошкоджень.

4) Не працюють світлодіоди:

- перевірити правильність підключення до Arduino;
- замінити несправний світлодіод.

5) Плата працює з перебоями:

- перевірити акумулятор;
- перевірити плату на наявність пошкоджень;

- перевірити плату на наявність пилу, в разі необхідності почистити;
- в разі необхідності замінити плату.

6) Не працює лампа освітлення:

- перевірити правильність підключення до ардуіно;
- перевірити чи працює реле;
- поміняти лампу в разі необхідності.

7) Не працює акумулятор:

- перевірити чи не надувся акумулятор;
- перевірити правильність підключення;
- замінити акумулятор в разі необхідності.

8) Не працює ключ:

- перевірити чи співпадає код ключа, з кодом вказаним в скетчі;
- поміняти ключ в разі необхідності, і ввести новий код ключа в скетч.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при пораненнях

Долікарська допомога при пораненнях, зв'язаних з платами, електричними елементами та іншими подібними ситуаціями, є важливим аспектом безпеки, який вимагає належного знання і умінь для ефективного надання допомоги потерпілому. В таких випадках перша допомога забезпечує важливий час для подальшого медичного втручання та мінімізувати ризик подальших ускладнень.

Одним із поширених видів поранень, пов'язаних з платами та електричними елементами, є ураження електричним струмом. У таких випадках першочерговим завданням надання допомоги є забезпечення безпеки для того хто допомагає та потерпілого. Найважливіше - відключити джерело електричного струму, переконавшись, що немає ризику ураження для себе. При цьому можна використовувати інструменти, що не проводять електричний струм (дерев'яні предмети або гумові рукавички) [13].

Після відключення джерела електричного струму слід провести оцінку стану потерпілого та викликати швидку медичну допомогу. У випадку, якщо потерпілий не дихає або серце його не б'ється, необхідно негайно розпочати проведення штучної вентиляції легень та кардіопульмональної реанімації (ШВЛ та КПР) до прибуття медичних працівників [14].

Якщо потерпілий не проявляє ознак дихання або серцебиття, але присутній електричний струм, необхідно відшукати спосіб безпечного відключення потерпілого від джерела струму, використовуючи безпечні матеріали, такі як дерев'яні рукоятки, гумові рукавички або ізольований металевий предмет. Для уникнення ураження струмом людини, що допомагає постраждалому, необхідно триматися подалі від мокрих поверхонь або металевих предметів.

Якщо внаслідок поранення виникла кровотеча, слід негайно прикласти тугий пов'язку на місце кровотечі, натискаючи на неї прямими рухами. При цьому можна використовувати чисті шматочки тканини або стерильні компреси. В разі наявності вогнища запалення слід накласти асептичний бинт на поранену ділянку, уникати використання металевих предметів або інструментів, що можуть призвести до проведення електричного струму [15].

Незалежно від типу поранення, потерпілого необхідно якомога швидше доставити до медичного закладу для подальшого обстеження та лікування. Важливо пам'ятати, що надання долікарської допомоги при пораненнях, пов'язаних з платами, електричними елементами та подібними ситуаціями, повинно проводитись лише кваліфікованими медичними працівниками або особами, які отримали відповідну підготовку.

Додатково важливою умовою є уникати будь-яких самостійних спроб видалення плат або електричних елементів з пораненої області. Це призводить до подальшого ушкодження тканин або виникнення додаткових ускладнень. Навіть якщо поранення здається незначним, важливо звернутися до медичних фахівців для оцінки ступеня ушкодження та прийняття відповідних заходів.

Крім того, слід пам'ятати про необхідність дотримання особистої безпеки під час надання долікарської допомоги. Носіть захисні рукавиці, щоб уникнути контакту з кров'ю або іншими тілесними рідинами, і уникайте безпосереднього контакту з електричними джерелами струму. Забезпечте належне освітлення та комфортне робоче середовище для оптимальної роботи.

У разі, якщо поранення зв'язане з електричними елементами, такими як акумулятори чи батареї, важливо уникати контакту з витіканням електроліту. Електроліт є корозивним і спричиняє подразнення шкіри або очей. У таких випадках слід негайно промити поранену ділянку водою протягом кількох хвилин та звернутися до медичних працівників для подальшої оцінки та лікування.

Необхідно пам'ятати, що надання долікарської допомоги при пораненнях, пов'язаних з платами, електричними елементами та іншими подібними ситуаціями, має бути проведено відповідно до місцевих медичних протоколів та рекомендацій. Кожен випадок вимагає індивідуального підходу, тому важливо діяти розумно та відповідально, звертаючись до професіоналів, коли це необхідно.

Надання долікарської допомоги при пораненнях, пов'язаних з платами, електричними елементами та подібними ситуаціями, допомагає зменшити ризик ускладнень та прискорити відновлення постраждалої особи. Найбезпечнішим та найефективнішим варіантом є негайне звернення до медичних фахівців та дотримання їх рекомендацій.

Фахівцям при роботі з сигналізацією необхідно знати і вміти надати долікарську допомогу при пораненнях.

4.2 Порядок надання домедичної допомоги постраждалим при раптовій зупинці серця

Раптова зупинка серця є катастрофічним медичним станом, що належить до найсерйозніших надзвичайних ситуацій, і потребує безвідкладних, енергійних та розумних дій для відновлення нормального кровообігу та ефективного функціонування серця. У цих випадках надання домедичної допомоги відіграє вирішальну роль у забезпеченні негайного реагування та порятунку постраждалої особи.

Важливо пам'ятати, що швидка реакція і негайне надання домедичної допомоги є життєво важливими при раптовій зупинці серця. У таких надзвичайних ситуаціях кожна секунда має вирішальне значення, оскільки кровообіг та подих постраждалої особи припиняються, що призводить до невідправлення кисню до мозку та інших органів, швидкого пошкодження клітин та неповоротних ускладнень.

Правильне виконання КПР (кардіопульмональна реанімація) в цих критичних моментах є надзвичайно важливим. Застосування компресій грудної клітки для підтримки кровообігу разом з штучною вентиляцією легень для забезпечення доставки кисню є ключовими компонентами КПР. Навички правильної техніки компресій та розташування рук, глибина та частота стисків, а також належне відкриття дихальних шляхів допоможуть забезпечити оптимальну реанімаційну дію.

Крім того, використання автоматизованого екстернального дефібрилятора (АЕД) може додатково підвищити шанси на виживання. АЕД є портативним пристроєм, який може аналізувати ритм серця та, у разі потреби, адмініструвати електричний шок для відновлення нормального ритму серця.

Ключовою складовою успішної реанімації є також координація з медичними працівниками та швидке повідомлення екстреної медичної служби. Це дозволяє забезпечити безперервну медичну допомогу та перехід постраждалої особи в лікарню для подальшого лікування та відновлення.

Необхідно підкреслити, що навички КПР та використання АЕД мають бути вивчені та практикуватися заздалегідь, оскільки правильна реакція у критичних ситуаціях може врятувати життя. Регулярне проходження навчань по наданню першої допомоги та оновлення знань стануть незамінними інструментами для громадськості, щоб бути готовими діяти в надзвичайних ситуаціях та допомагати тим, хто опинився у важкому стані раптової зупинки серця.

Фахівці при роботі з сигналізацією мають знати основи домедичної допомоги, включаючи КПР та використання АЕД, ці знання є важливими оскільки вони дозволяють бути готовими реагувати в надзвичайних ситуаціях та надати необхідну допомогу, поки не прибуде швидка [14].

ВИСНОВКИ

У результаті виконання даної кваліфікаційної роботи було проведено розробку комплексної системи сигналізації з дистанційним керуванням на базі мікропроцесорної платформи Arduino UNO. Проект включав в себе аналіз, проектування, розробку та тестування системи з урахуванням актуальності теми та існуючих рішень у цій галузі.

У загальній частині проекту було проведено обґрунтування актуальності теми кваліфікаційної роботи, виявлено необхідність розробки нової системи сигналізації з дистанційним керуванням. Проведений аналітичний огляд існуючих рішень дозволив зрозуміти переваги та недоліки наявних систем, що послужило основою для подальшої розробки власного проекту.

У розділі розробки технічного та робочого проекту було виконано аналіз технічного завдання кваліфікаційної роботи, що дозволило чітко сформулювати вимоги до системи сигналізації та розробити структурну схему пристрою. Було проведено детальний опис і обґрунтування вибору елементної бази, зазначено ключові параметри та властивості обраної платформи Arduino UNO. Також була розроблена функціональна схема системи, що визначила взаємозв'язок між компонентами і їх роль у роботі системи. На основі цього був розроблений алгоритм роботи системи, який забезпечує правильну послідовність дій і взаємодію всіх компонентів.

У розділі налаштування сигналізації була розроблена інструкція з експлуатації електронного пристрою, що містить детальні вказівки щодо встановлення, налаштування та використання системи сигналізації. Також була розроблена методика перевірки, функціонування та контролю електронного пристрою, що дозволяє забезпечити його надійну роботу та виявити можливі несправності.

Основи охорони праці та безпеки життєдіяльності були ретельно розглянуті. Були розроблені правила надання долікарської допомоги при

пораненнях, що забезпечують безпеку користувачів системи. Також були встановлені процедури надання допомоги в надзвичайних ситуаціях, зокрема при раптовій зупинці серця.

В результаті виконання даної кваліфікаційної роботи було отримано значний практичний досвід у розробці комплексних систем, використанні мікропроцесорних платформ та програмуванні. Результатом роботи є функціональна система сигналізації з дистанційним керуванням, яка може бути успішно використана у різних сферах та застосуваннях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. StarterKit – Стартовий комплект системи безпеки Ajax – URL: <https://ajax.systems/ua/products/starterkit/>
2. Беспроводная GSM + WiFi сигнализация Smart 103 (PG-103) – URL: <https://bezpeka.top/besprovodnaja-gsm-wifi-signalizatsija-smart103>
3. Структурна схема – URL: https://uk.wikipedia.org/wiki/Структурна_схема
4. Обзор платы Arduino Uno Atmega328P – URL: <https://kolotushkin.com/1arduinouno.php>
5. Arduino – URL: <https://arduino.ua/?searchstring=Arduino+Uno>
6. ІЧ датчик руху на Arduino HC-SR501 – URL: <https://arduino.ua/prod193-ik-datchik-dvijeniya-dlya-arduino-hc-sr501>
7. GSM модуль на SIM800L – URL: <https://arduino.ua/ru/prod1665-gsm-modyl-na-sim800l>
8. Модуль 2 реле 5В з опторозв’язкою – URL: <https://arduino.ua/prod5868-modyl-2-rele-5v-z-optorazvyazkoi>
9. Зарядний модуль TP4056 Mini-USB – URL: <https://arduino.ua/prod3770-zaryadnii-modyl-tp4056-mini-usb-s-fynkciei-zashhiti-akkymylyatora>
10. Функційна електрична схема – URL: https://uk.wikipedia.org/wiki/Функційна_електрична_схема
11. Kolotushkin – URL: <https://kolotushkin.com/article.php?id=36>
12. База электронных компонентов – URL: <https://alexgyver.ru/lessons/components/>
13. Порядок надання домедичної допомоги постраждалим при ураженні електричним струмом – URL: <https://zakon.rada.gov.ua/laws/show/z0356-22#n800>
14. Порядок надання домедичної допомоги постраждалим при рпатовій зупинці серця – URL: <https://zakon.rada.gov.ua/laws/show/z0750-14#n41>

15. Порядок надання домедичної допомоги постраждалим при зовнішній кровотечі – URL: <https://zakon.rada.gov.ua/laws/show/z0356-22#Text>
16. Cyberforum – URL: <http://www.cyberforum.ru/cpp-beginners/>
17. Мікропроцесори та мікропроцесорні комплекти інтегральних мікросхем. Довідник, під ред. В.А. Шахнова, том 2, М., Радио и связь, 1988. - 156 с.
18. Микропроцессоры. Учебное пособие в 5-ти книгах, под редакцией В.А. Шахнова, Москва Высшая школа, 1988. - 96 с.
19. Грибан В. Г., Негодченко О. В. Охорона праці : навчальний посібник . - 2-е видання. Київ: Центр учбової літератури, 2018.- 280 с, ISBN 978-966-364-832-3
20. Запорожець О. І., Протоєрейський О. С., Франчук Г. М., Боровик І. М. Основи охорони праці підручник Київ: Центр учбової літератури, 2017. - с.264, ISBN 978-617-673-423-9
21. М. С. Одарченко, А. М. Одарченко, В. І. Степанов, Я. М. Черненко. Основи охорони праці: підручник/ – Х. : Стиль-Издат, 2017. – 334 с. ISBN 966-7885-84-4
22. Основи охорони праці: Підруч для студ вищих навч закладів За ред мп Гандзюка - К Каравела, 2004 - 408 с.
23. Охорона праці в галузі комп'ютерингу: підручник / Л. А. Катренко, А. В. Катренко ; [за наук. ред. В. В. Пасічника] ; М-во освіти і науки, молоді та спорту України. — Л. : Магнолія 2006, 2012. — 544 с

ДОДАТОК А

```
#include <OneWire.h> //бібліотека для зчитування ключів
#include "EEPROM.h" //бібліотека для запису в енергозалежну
пам'ять

#include <SoftwareSerial.h> //бібліотека для створення другого
програмного UART для зв'язку с GSM модулем

const int key_led = 5; //Пін 5 для підключення світлодіода для
індикації про постанову на охорону

const int key_pin = 6; //Пін 6 для зчитування ключа таблетки
RW1990, TM1990A

OneWire key (key_pin);

const int rx_prog = 2; //програмний rx для підключення GSM
модуля SIM800L

const int tx_prog = 3; //програмний tx для підключення GSM модуля
SIM800L

SoftwareSerial SIM800(rx_prog, tx_prog); //програмні RX, TX для
зв'язку з модулем SIM800L

const int led_2 = 9; //світлодіод для індикації сигналізації
const int a1 = 15; //лінія A1 для підключення датчика
const int a2 = 16; // лінія A2 для підключення датчика
const int a3 = 17; // лінія A3 для підключення датчика
const int a4 = 18; // лінія A4 для підключення датчика
const int a5 = 19; // лінія A5 для підключення датчика
const int speaker = 8; //контакт для підключення сирени
const int set_220 = 7; //контакт для визначення наявності живлення
в мережі

const int relay = 13; //контакт для підключення твєдротілого реле
або релейного модуля
```

```

String phon = "+38097xxxxxxx";
//1 номер на який буде здійснюватися дзвінок якщо спрацює
сигналізація

String phones = "+38096xxxxxxx, +38067xxxxxxx, +48610xxxxxxx";
//1 або декілька номерів через кому, з котрих будет дозволено
керування по СМС

String balance = "*111#";
//1 номер для перевірки балансу Sim карти GSM сигналізації

String SMS_phone = "+38097xxxxxxx";
//1 номер на який будуть приходити SMS звіти

String sms_otchet = "Ktr3"; //запит SMS про стан сигналізації,
датчиків сигналізації і реле

String sig_on = "Sig1"; //поставити сигналізацію на охорону
String sig_off = "Sig0"; //зняти сигналізацію з охорони
String sms_on = "Sms1"; //ввімкнути відправку SMS при
спрацюванні сигналізації
String sms_off = "Sms0"; //вимкнути відправку SMS при
спрацюванні сигналізації
String ring_on = "Ring1"; //ввімкнути вихідний виклик при
спрацюванні сигналізації
String ring_off = "Ring0"; //вимкнути вихідний виклик при
спрацюванні сигналізації
String ballanse = "Bal"; //перевірка балансу SIM карти
String relay_on = "Rel1"; //команда для включення реле
String relay_off = "Rel0"; //команда для виключення реле

```



```

byte arr;
String read_key = "";

String open_key = "1f34bf6000f7";
//1 дозволений код ключ (для додавання декількох ключів коди
записуються через кому)

int sig_pause = 10; //10 секунд пауза після зчитування ключа при
постановці на охорону
int block_pause = 10; //10 секунд пауза для повторного зчитування
ключа, після того як був зчитаний невірний ключ
long spiker_time = 20; //20 секунд час роботи сирени, після
відключення датчиків
long ring_time = 20; //20 секунд пауза між дзвінками при
спрацьовуванні сигналізації
int batery_limit = 50; //значення заряду батареї в відсотках, при якому
відправиться СМС про розряджену батарею (оптимально від 30% до 60%)
int ring_x = 3; //пауза 3 хвилини між серіями дзвінків, при
постійному спрацьовуванні кінцевика (від 1 до 32000 хвилин)

int val, send_mesag_state_Batery = 0, send_mesag_Balanse = 0,
led_2_OFF, led_2_ON, readflag = 0;
int sig_st = 0, stat_a1 = 0, stat_a2 = 0, stat_a3 = 0, stat_a4 = 0, stat_a5 = 0;
int ring = 0, ring_st = 1, ring_fl = 0, ring_fl2 = 0, send_mesag = 0, sms_st
= 1, blinker = 1, rel_st = 0;
String signalingS = "", signaling = "", str_a1 = "", str_a2 = "", str_a3 = "",
str_a4 = "", str_a5 = "";
String signaling_OK = "Signaling OK";

```

```

int k=0, v=0, t=0, msgphone = 0, state_Batery = 100, flag_Button_KeY =
1, flag_Button_KeY_P = 0;

// Змінні для роботи з EEPROM
int flag_EEPROM, start_FLAG = 2, address_FLAG = 0, address_SMS =
2, address_SIG = 4, address_RING = 6, address_REL = 8;

// Змінні для зберігання точок відліку, для таймерів
unsigned long timing10, timing11, timing12, timing13, timing14, timing15,
timing16, timing17;

String _response = ""; // Змінна для зберігання
відповіді модуля
long lastUpdate = millis(); // Час останнього оновлення
long updatePeriod = 60000; // Перевіряти кожную
хвилину

String waitResponse() { // Функція очікування
відповіді і повернення отриманого результату
String _resp = ""; // Зміна для зберігання
результату
long _timeout = millis() + 10000; // Змінна для
відстеження тайм ауту (10 секунд)
while (!SIM800.available() && millis() < _timeout) {}; // Чекаємо
відповіді 10 секунд, якщо прийшла відповідь або настав тайм аут, то...
if (SIM800.available()) { // Якщо є, що зчитувати...
_resp = SIM800.readString(); // зчитуємо і
запам'ятовуємо
}else { // Якщо прийшов тайм аут, то...
Serial.println("Timeout..."); // сповіщаємо про це і...

```

```

        }return _resp; // повертаємо результат. Пусто,
якщо проблема
    }

    String sendATCommand(String cmd, bool waiting) {
        String _resp = ""; // Змінна для зберігання
результату
        Serial.println(cmd); // Дублюємо команду в
монітор порту
        SIM800.println(cmd); // Відправляємо команду
модулю
        if (waiting) { // Якщо необхідно дочекатись
відповіді...
            _resp = waitResponse(); // ... чекаємо, коли буде
передана відповідь
            // Якщо Echo Mode вимкнений (ATE0), то ці 3 строки можна
закоментувати
            if (_resp.startsWith(cmd)) { // Прибираємо з відповіді
дубльовану команду
                _resp = _resp.substring(_resp.indexOf("\r", cmd.length()) + 2);}
            Serial.println(_resp); // Дублюємо відповідь в
монітор порту
        }return _resp; // Повертаємо результат. Пусто,
якщо проблема
    }

    void parseSMS(String msg) { // Парсимо SMS
        String msgheader = "";
        String msgbody = "";

```



```
EEPROM.put(address_SMS, sms_st);} //вимкнути відправку SMS  
при спрацьовуванні сигналізації
```

```
if(msgbody == sms_on){ Serial.println(sms_on); delay(300); sms_st = 1;  
EEPROM.put(address_SMS, sms_st);} //ввімкнути відправку SMS  
при спрацьовуванні сигналізації
```

```
if(msgbody == ring_off){ Serial.println(ring_off); delay(300); ring_st = 0;  
EEPROM.put(address_RING, ring_st);} //вимкнути вихідний виклик  
при спрацьовуванні сигналізації
```

```
if(msgbody == ring_on){ Serial.println(ring_on); delay(300);ring_st =  
1;ring = 0;  
EEPROM.put(address_RING, ring_st);} //ввімкнути вихідний виклик  
при спрацьовуванні сигналізації
```

```
if(msgbody == sig_off){ Serial.println(sig_off); delay(300);sig_st = 0;  
EEPROM.put(address_SIG, sig_st);} //вимкнути сигналізацію
```

```
if(msgbody == sig_on){ Serial.println(sig_on); delay(300); sig_st = 1;  
EEPROM.put(address_SIG, sig_st);} //ввімкнути сигналізацію
```

```
if(msgbody == relay_off){ Serial.println(relay_off); delay(300);rel_st = 0;  
EEPROM.put(address_REL, rel_st);} digitalWrite(relay, rel_st);  
//вимкнути реле
```

```
if(msgbody == relay_on){ Serial.println(relay_on); delay(300); rel_st = 1;  
EEPROM.put(address_REL, rel_st);} digitalWrite(relay, rel_st);  
//ввімкнути реле
```

```
if(msgbody == ballanse){ Serial.println(ballanse); //перевірка балансу
```

SIM карти

```
    SIM800.println("AT+CUSD=1,\"\" + balance + "\"");}
```

```
  }else {
```

```
//Serial.println("Unknown phonenumber");
```

```
}}
```

```
void numbr_sms(){
```

```
    SIM800.println("AT+CMGS=\"\" + SMS_phone + "\"");
```

```
    delay(3000);
```

```
}
```

```
void text_sms(){
```

```
SIM800.println("Set = " + String(v) + " volt");delay(300);
```

```
if (state_Batery == 10){state_Batery = 100;}
```

```
SIM800.println("Battery = " + String(state_Batery) + "%");delay(300);
```

```
SIM800.println(signaling);delay(300);
```

```
SIM800.println(signalingS);delay(300);
```

```
SIM800.println(str_a1);delay(300);
```

```
SIM800.println(str_a2);delay(300);
```

```
SIM800.println(str_a3);delay(300);
```

```
SIM800.println(str_a4);delay(300);
```

```
SIM800.println(str_a5);delay(300);
```

```
SIM800.println("Ring = " + String(ring_st)); delay(300);
```

```
SIM800.println("SMS = " + String(sms_st)); delay(300);
```

```
SIM800.println("REL = " + String(rel_st));
```

```
}
```

```
void go_sms(){
```

```
    delay(300);
```



```

EEPROM.get(address_FLAG, flag_EEPROM);
if(flag_EEPROM < start_FLAG){
    EEPROM.put(address_FLAG, start_FLAG);
    EEPROM.put(address_SMS, sms_st);
    EEPROM.put(address_SIG, sig_st);
    EEPROM.put(address_RING, ring_st);
    EEPROM.put(address_REL, rel_st);
}
EEPROM.get(address_SMS, sms_st);
EEPROM.get(address_SIG, sig_st);
EEPROM.get(address_RING, ring_st);
EEPROM.get(address_REL, rel_st);

Serial.begin(9600);                // Швидкість обміну даними з
комп'ютером
SIM800.begin(9600);                // Швидкість обміну даними з
модемом
analogReference(INTERNAL);

sendATCommand("AT", true);        // Відправили AT для
налаштування швидкості обміну даними
sendATCommand("AT+CMGDA=\"DEL ALL\"", true); // Видаляємо всі
SMS, щоб не забивати пам'ять
sendATCommand("AT+CMGF=1;&W", true); // Вмикаємо
текстовий режим SMS (Text mode) й одразу зберігаємо значення (AT&W)!
//sendATCommand("AT+CSCB=1;&W", true); // Вмикаємо прийом
спеціальних повідомлень й одразу зберігаємо значення (AT&W)!
lastUpdate = millis();            // Скидаємо таймер

pinMode(set_220, INPUT);

```



```

        if(val == '7'){ring_st = 1; ring = 0;} //ввімкнути вихідний виклик при
спрацьовуванні сигналізації
        if(val == '8'){ring_st = 0;} //вимкнути вихідний виклик при
спрацьовуванні сигналізації
        if(val == '9'){sms_st = 1;} //ввімкнути відправку SMS при
спрацьовуванні сигналізації
        if(val == '0'){sms_st = 0;} //вимкнути відправку SMS при
спрацьовуванні сигналізації

    }

    if (lastUpdate + updatePeriod < millis() ) { // Перевірка
наявності нових повідомлень
        do {
            _response = sendATCommand("AT+CMGL=\"REC UNREAD\",1",
true);// Відправляємо запит зчитування непрочитаних повідомлень
            if (_response.indexOf("+CMGL: ") > -1) { // Якщо є хоч одне,
отримуємо його індекс
                // Serial.println ("POLUCHENO 1 SMS");
                int msgIndex = _response.substring(_response.indexOf("+CMGL: ") + 7,
_response.indexOf("\"REC UNREAD\"", _response.indexOf("+CMGL: ")) -
1).toInt();
                char i = 0; // Оголошуємо лічильник спроб
                do { i++; // Збільшуємо лічильник
                    _response = sendATCommand("AT+CMGR=" + (String)msgIndex + ",1",
true); // Намагаємося отримати текст SMS по індексу
                    _response.trim(); // Прибираємо пробіли на
початку/кінці
                    if (_response.endsWith("OK")) { // Якщо відповідь
закінчується на "OK"

```

```

        if (!hasmsg) hasmsg = true;                // Ставимо флаг
нааявності повідомлень для видалення
        sendATCommand("AT+CMGR=" + (String)msgIndex, true);    //
Робимо повідомлення прочитаним
        sendATCommand("\n", true);                // Перестраховка –
вивід нового рядка
        parseSMS(_response);                      // Відправляємо текст
повідомлення на обробку
        break;                                    // Вихід з do{ }
    }else {                                        // Якщо повідомлення не
закінчується на ОК
        // Serial.println ("Error answer");        // Якась помилка
        sendATCommand("\n", true);                // Відправляємо новий
рядок і повторюємо спробу
    }} while (i < 10);
    break;}
    else {
        lastUpdate = millis();                    // Скидаємо таймер
        if (hasmsg) {
            sendATCommand("AT+CMGDA=\"DEL READ\"", true);    //
Видаляємо всі прочитані повідомлення
            hasmsg = false;
            }break;}} while (1);}

    if (SIM800.available()) {                    // Якщо модем щось
відправив...
        _response = waitResponse();              // Отримуємо відповідь від
модему для аналізу
        _response.trim();                        // Прибираємо зайві пробіли на
початку і в кінці

```

```

        Serial.println(_response);           // Якщо потрібно виводимо в
монітор порту

        if (_response.indexOf("+CMTI:")>-1) {           // Прийшло
повідомлення про відправку SMS
            lastUpdate = millis() - updatePeriod;     // Тепер немає необхідності
обробляти SMS тут, достатньо просто
        }

        if (_response.indexOf("+CBC:")>-1) {           //Якщо прийшла
відповідь про стан батареї
            state_Batery = _response.substring(17, 19).toInt(); //Висмикуємо частину
рядка з даними про стан батареї, переводимо в integer

            /*Закоментовані рядки з функціями Serial.print або Serial.println
служать для більш зручного налаштування по UART.
Щоб спостерігати всі процеси в моніторі порту, необхідно
закоментувати всі подібні рядки що знаходяться в всьому void loop()*/
            //Serial.print("Batery = "); Serial.println(state_Batery);
            //Serial.print("Set = "); Serial.println(v);
        }

        //команды для перевірки балансу
        if (_response.indexOf("+CUSD:")>-1) {           //Якщо прийшла
відповідь про стан балансу
            //Serial.println(">>Balanse!<<");
            send_mesag_Balanse++;
        }

        if (k == 3){delay(300);

```

```

// Serial.println("<<Send SMS (Ktr3)>>");
numbr_sms(); text_sms(); go_sms();k=0;}

//опитування датчиків
stat_a1 = digitalRead(a1);
if (stat_a1 == LOW && signalingS == signaling_OK) { str_a1 = "A1
OK!";}
if (stat_a1 == HIGH) {str_a1 = "A1 OPEN!";}

stat_a2 = digitalRead(a2);
if (stat_a2 == LOW && signalingS == signaling_OK) { str_a2 = "A2
OK!";}
if (stat_a2 == HIGH) {str_a2 = "A2 OPEN!";}

stat_a3 = digitalRead(a3);
if (stat_a3 == LOW && signalingS == signaling_OK) { str_a3 = "A3
OK!";}
if (stat_a3 == HIGH) {str_a3 = "A3 OPEN!";}

stat_a4 = digitalRead(a4);
if (stat_a4 == LOW && signalingS == signaling_OK) { str_a4 = "A4
OK!";}
if (stat_a4 == HIGH) {str_a4 = "A4 OPEN!";}

stat_a5 = digitalRead(a5);
if (stat_a5 == LOW && signalingS == signaling_OK) { str_a5 = "A5
OK!";}
if (stat_a5 == HIGH) {str_a5 = "A5 OPEN!";}

//Сигналізація виключена

```

```

if(sig_st == 0){
    send_mesag = 0; ring = 0; ring_fl = 0;
    signaling = "Signal OFF"; digitalWrite(speaker, LOW);}

//Сигналізація включена
if(sig_st == 1){  signaling = "Signal ON";

// якщо датчик спрацює
if (ring == 0 && led_2_ON == 200 && ring_fl == 0)
    { signalingS = "ПОПУТКА VZLOMA"; digitalWrite(speaker, HIGH);
if (ring_st == 1){ring ++; Serial.println("<<ring ++;>>");}
if(sms_st == 1){send_mesag++; }
    timing10 = millis(); Serial.println("VZLOM");
    } else { signalingS = signaling_OK;
    if (millis() - timing12 > spiker_time){timing12 = millis();
digitalWrite(speaker, LOW);}
    }
}

// 3 дзвінка при спрацьовуванні датчиків
if (ring > 0 && millis() - timing10 > ring_time * 1000 && ring_st == 1
&& ring_fl == 0){
    timing10 = millis(); ring++;
    String ring1 = String(ring-1);
    Serial.println("<< Ring " + ring1 + " >>");
    SIM800.println("ATD" + phon + "");
}

if (ring > 3) {ring = 0; send_mesag = 0; ring_fl = 1;}

```

```

//опитування датчиків
if(stat_a1 == LOW && stat_a2 == LOW && stat_a3 == LOW && stat_a4
== LOW && stat_a5 == LOW)
{blinker = 1;}else{blinker = 0;}

//блінк для індикації сигналізації
if(sig_st == 0 && blinker == 1){led_2_ON = 1000; led_2_OFF = 1000;}
if(sig_st == 1 && blinker == 1){led_2_ON = 3000; led_2_OFF = 100;}

//опитування датчиків
if(stat_a1 == HIGH || stat_a2 == HIGH || stat_a3 == HIGH || stat_a4 ==
HIGH || stat_a5 == HIGH)
{led_2_ON = 200; led_2_OFF = 100;
if(sig_st == 1){digitalWrite(speaker, HIGH);}
}else {
if (millis() - timing12 > spiker_time*1000){timing12 = millis();
digitalWrite(speaker, LOW);}
}

if (millis() - timing13 > led_2_ON){
timing13 = millis();timing14 = millis();
digitalWrite(led_2, HIGH);
}
if (millis() - timing14 > led_2_OFF){digitalWrite(led_2, LOW);}

if (send_mesag == 1 && sms_st == 1){
// Serial.println("<<<Soobshenie Otpravleno (Pri vzlome)>>>");
numbr_sms(); text_sms(); go_sms(); send_mesag++;
}

```

```

if (send_mesag > 2) {send_mesag = 2;}

if (send_mesag_Balanse == 1){
  // Serial.println("<<SMS Balanse>>");
  numbr_sms(); SIM800.println(_response); go_sms();
  send_mesag_Balanse = 0; delay(1000);
  sendATCommand("AT+CMGDA=\"DEL ALL\"", true);// Видаляємо всі
SMS, щоб економити пам'ять
}
if (send_mesag_Balanse >= 2) {send_mesag_Balanse = 0;}

//Повідомлення про розряджену батарею
if(state_Batery > 11 && state_Batery <
battery_limit){send_mesag_state_Batery++;}
if (send_mesag_state_Batery == 1){
  // Serial.println("<<SMS Batery < 50%>>");
  numbr_sms();
  SIM800.println("Set = " + String(v) + " volt");
  delay(300);
  SIM800.print("Battery = " + String(state_Batery) + "%");
  go_sms();
}
if (send_mesag_state_Batery > 2) {send_mesag_state_Batery = 2;}

if(state_Batery > 80){send_mesag_state_Batery = 0;}

int sensorVal = digitalRead(set_220); // для отримання стану мережі
220
if (sensorVal == HIGH) { v = 220;} else {v = 0;}

```



```

if (millis() - timing11 > 1000){
    timing11 = millis();
    read_Button_KeY();           // зчитування ключа і якщо ключ
    вірний то змінити стан сигналізації

    if(t == 60){
        SIM800.println("AT+CBC");      // Отримати рівень заряду батареї
        //Serial.print ("Ring = " + String(ring_st));
        //Serial.print ("SMS = " + String(sms_st));
        //Serial.println ("STAT BAT?");

        if (ring_fl == 1){ring_fl2++;}
        if (ring_fl2 == ring_x+1){ring_fl = 0; ring_fl2 = 0;}
        }t++; if(t > 60){t = 0;}
    }

    //постановка на охорону ключем таблеткою
    if(sig_st == 0 && flag_Button_KeY_P == 1 && millis() - timing15 >
sig_pause * 1000){
        sig_st = 1; flag_Button_KeY_P = 0; //затримка спрацьовування після
зчитування ключа, виставляється змінною sig_pause на початку скетчу
        flag_Button_KeY = 1;           //дозволяємо зчитування ключа
        EEPROM.put(address_SIG, sig_st); //зберігаємо стан в
енергонезалежну пам'ять
        digitalWrite(key_led, LOW);
    }

    //зняття з охорони ключем таблеткою
    if(sig_st == 1 && flag_Button_KeY_P == 1){
        sig_st = 0; flag_Button_KeY_P = 0; timing16 = millis();
    }

```

```

EEPROM.put(address_SIG, sig_st); //зберігаємо стан в
енергонезалежну пам'ять
    digitalWrite(key_led, LOW);
}

if(sig_st == 0 && flag_Button_KeY_P == 0 && flag_Button_KeY == 1
&& millis() - timing16 > 3000){
    flag_Button_KeY = 1;          //через 3 секунди після зняття з
охорони дозволяємо зчитування ключа
}

//при невірному коді ключа не дозволяємо зчитувати ключ впродовж
часу block_pause
if(flag_Button_KeY == 0 && millis() - timing17 > block_pause * 1000){
    flag_Button_KeY = 1;}        //дозволити сканування ключа після
закінчення часу block_pause

}

```