**РОЗДІЛ 3**

**ПРОГРАМНА ТА АПАРАТНА РЕАЛІЗАЦІЯ ЗАПРОПОНОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТИПОЖЕЖНОЇ БЕЗПЕКИ**

3.1 Принцип роботи *GSM* модуля

Після підключення модуля *SIM800L* до плати *Аrduino Nano*, він починає шукати мобільну мережу та автоматично реєструється. За допомогою вбудованого світлодіода дізнаємося про стан модуля (без знайденої мобільної мережі – швидке мигання, мобільна мережа знайдена – повільне блимання). [20] На рис. 3.1 зображено *GSM* – модуль *SIM800L*.

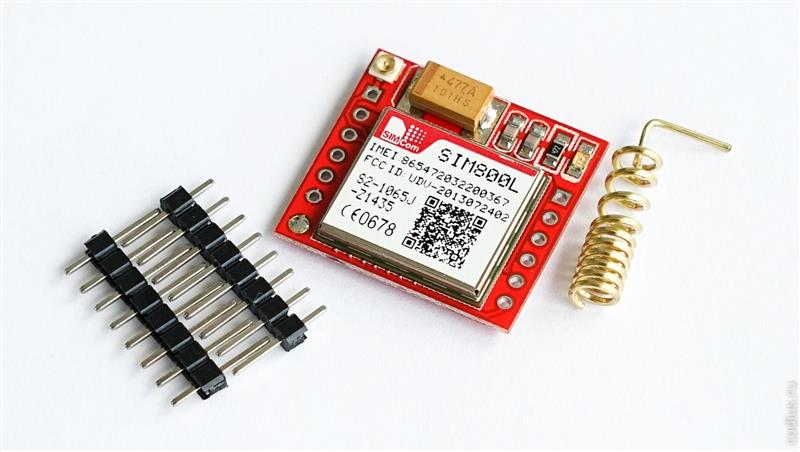


Рис.3.1 — *GSM*–модуль *SIM800L*

Сам *GSM* модуль *SIM800L* знаходиться за металевою кришкою.

«Сердцем» *GSM* модуля *SIM800L* являється чіп *Mediatek ARM MT6261*. Для *GSM / GPRS* – зв’язку задає 4 – діапазонний (*GSM850 / GSM900 / DCS1800 / PCS1900*)приймач *RF7198*. [20] На рис. 3.2 справа на зеленій друкованій платі зображено чіп модуля *SIM800L*. [20]



Рис.3.2 — Чіп *GSM*–модуля *SIM800L*

*MediaTek MT6261* – це така однокристальна система, яка призначена для першої черги для носіїв електроніки, в основному для розумних годиників, Включає до себе: *Bluetooth, LCD, GSM + GPRS*, камеру і *FM*-радіо. На рис. 3.3 зображено чіп *Mediatek ARM MT6261* та приймач *RF7198*. [19]

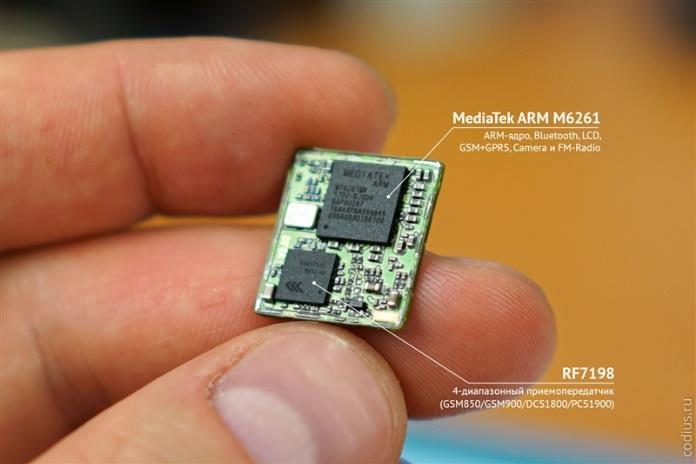


Рис.3.3 — Чіп *Mediatek ARM MT6261* та приймач *RF7198*

З вище сказаного зрозуміло, що різні модулі серії 800 мають різну характеристику і для остаточного вибору треба провести аналіз їх характеристик і функціоналу. Якщо відкинути несуттєві і однакові для всіх модулів характеристики, то принципові відмінності можна коротко представити у вигляді таблиці. Всі модулі мають доволі незначні відмінності один з одним. Нам не важлива наявність *FM* – приймача та *Bluetooth*, то в принципі можна обрати любий із цієї серії. В даній охоронній сигналізації будемо використовувати *GSM* модуль *SIM800L*. [20] В табл. 3.1 вказана порівняльна характеристика модулів серії 800. [20]

Таблиця 3.1 — Характеристика модулів серії 800.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **SIM800** | **SIM800C** | **SIM800H** | **SIM800L** | **SIM800F** | **SIM868** | **SIM808** |
| Голосові  виклики | + | + | + | + | + |  |  |
| GNSS (GPS  ) |  |  |  |  |  | + | + |
| Bluetooth  3.0 | + | + | + |  |  |  |  |
| Вбудований FM-  приймач: (76-109  МГц) з кроком 50  кГц |  |  | + | + |  |  |  |
| CSD | + |  | + | + |  |  | + |

*Початок роботи з модулем SIM800L*

Щоб почати роботу з модулем *SIM800L* ознайомимося з його функціональною схемою та розпіновкою контактів. Щоб розпочати повноцінну роботу с модулем треба придбати робочу *SIM* – карта формату *micro SIM.* На рис. 3.4 зображена функціональна схема *GSM* модуля *SIM800L*.

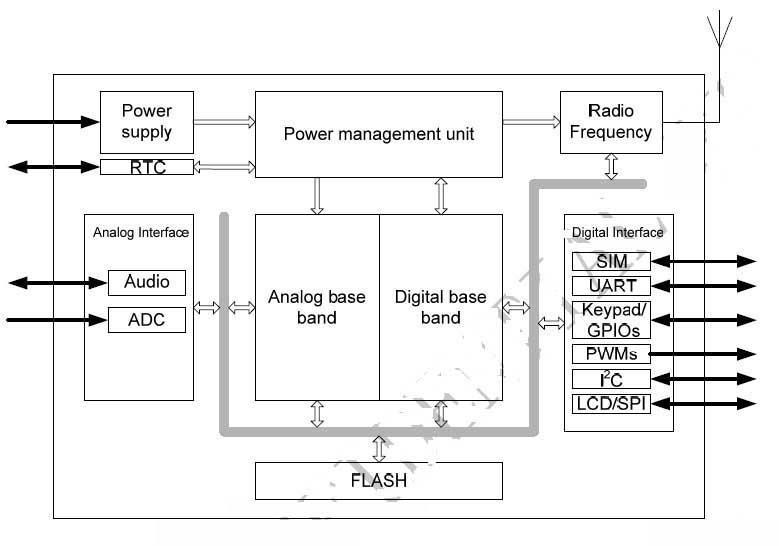


Рис.3.4 — Функціональна схема *GSM* модуля *SIM800L*

На рис. 3.5 зображено розпіновку *GSM* модуля *SIM800L*.

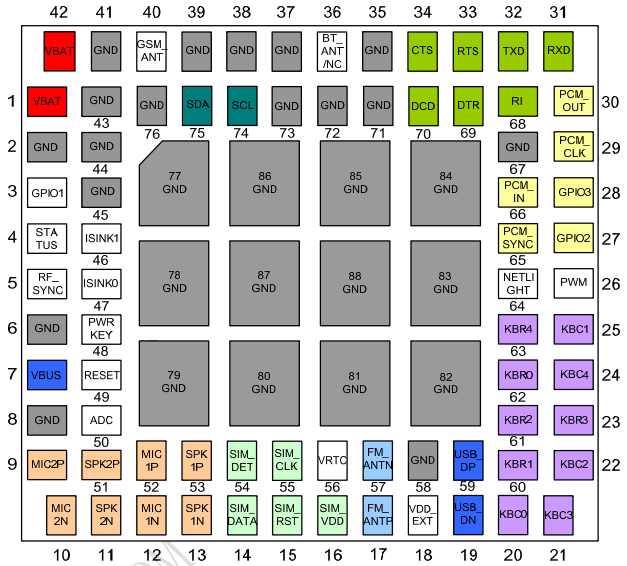


Рис.3.5 — Розпіновка *GSM* модуля *SIM800L*

Живлення модуля здійснюється напругою в діапазоні 3,7 – 4 В. Дана напруга є нестандартною напругою для плати *Arduino Nano.* Живити модуль можна двома способами від зовнішнього джерела живлення або використовувати понижуючий перетворювач напруги.

Не можна *GSM* модуль *SIM800L* живити від плати *Arduino*. Тому що модуль буває доволі «ненажерливим», його максимальне споживання струму досягає 2 А. Як уже було розглянуто плата *Arduino* не в змозі забезпечити його таким струмом. Скоріш за все, якщо спробувати підключити напряму до плати *Arduino*, то вірогідний збій в роботі, або щось вийде з ладу, або буде працювати некоректно. На рис. 3.6 показано два варіанти рішення проблеми живлення модуля. [20]

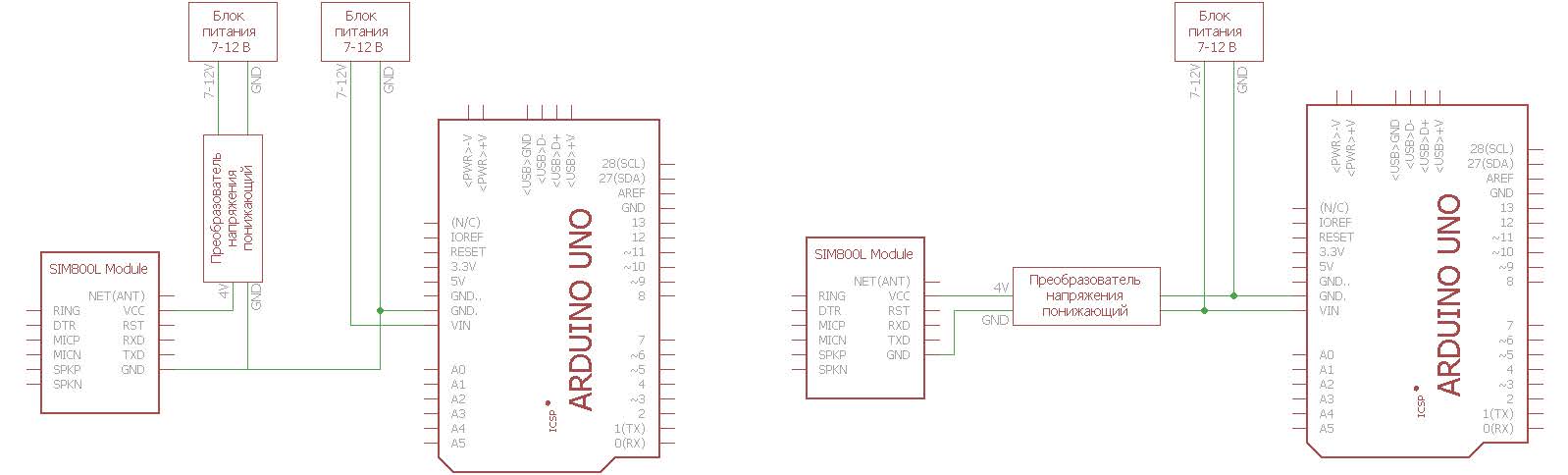


Рис.3.6 — Живлення модуля

*Підключення GSM модуля SIM800L до Arduino Nano*

Як і живлення, у модему не стандартний логічний рівень 2,8 В. При спробі підключити що до 5 або 3,3 вольта, модем буде вимикатися.

Тому наступним кроком після налагодження живлення модуля необхідно підключити відповідні контакти до контактів плати *Arduino*. Підключати напряму контакти *RX, TX* не можна. В описі модуля сказано про максимальний рівень логічної одиниці на вході *RX* яка становить 3,1 В, мінімальна 2,1 В. На рис. 3.7 вказані характеристики послідовного порту.

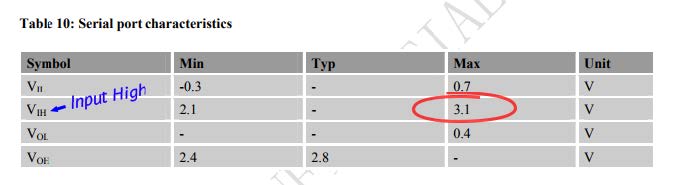


Рис.3.7 — Характеристики послідовного порту

На платі *Arduino Nano* відсутні елементи, що допоможуть конвертувати вхідну напруги навантаження. Тому можна використовувати дільник напруги.

Для того щоб організувати потрібний рівень логічної одиниці номіналу 2,5 В необхідно взяти два однакових за номіналом резистора в діапазоні 1 – 10 КОм. На рис.3.8 наведений приклад дільника напруги та схеми підключення *GSM* модуля к *Arduino Uno* із використанням дільника напруги.

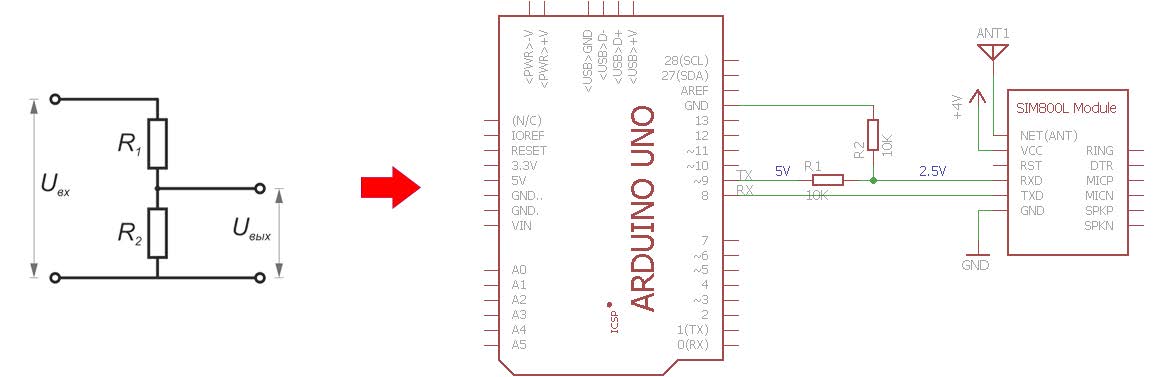


Рис.3.8 — Схема підключення *GSM* модуля до *Arduino Uno* із використанням дільника напруги

Контакт, на платі *Arduino Nano*, *RX* знаходиться в режимі «ВХОДУ», та повністю нормально реагує на логічну одиницю *GSM* модуля, яка становить 2,8 В. [20]

3.2 Вибір мови програмування

Для програмування приладів *Arduino* використовуються мови *Assembler, C і C++.* Нам потрібна мова програмування, яка може компілюватися і використовуватись ефективно. Тобто переводитися нулі і одиниці якомога оптимальніше, без витрат дорогоцінних комірок пам'яті. Використовуючи мови *Assembler, C і C++* навіть у вузьких рамках ресурсів мікроконтролера, можна писати багаті можливостями програми, які працюють досить швидко. *Assembler*, не можна назвати простим, тобто, фаворитом для *Arduino* є *C/C++.*

Мова програмування *C++* — це надбудова над мовою програмування *C*. Будь-яка програма на *C* є коректною програмою для *C++,* але не навпаки. Але є можливість користуватися і тим і іншим. В даній кваліфікаційній роботі була обрана мова програмування *C++,* це обумовлено тим, що мова *C++* не вимагає високої кваліфікації програміста та легко прошивається через кросплатформний додаток *Arduino IDE (integrated development environment*).

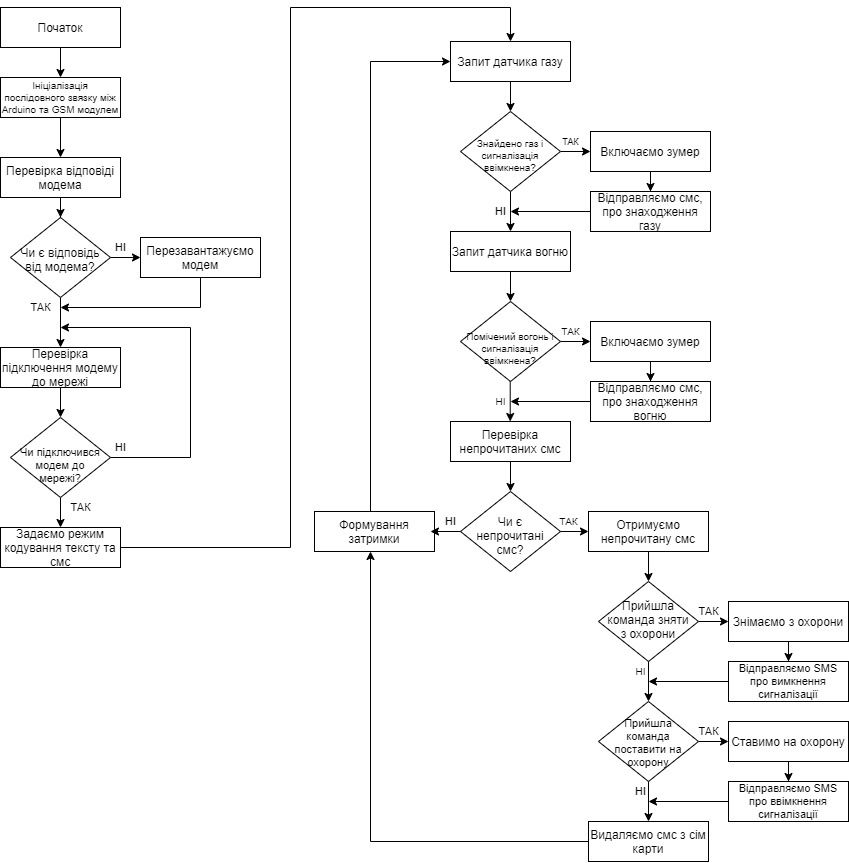
3.3 Розробка блок-схеми алгоритму програми

Розроблювану програму можна розділити на 3 частини:

* ініціалізація МК
* зчитування показників з датчиків;
* перевірка новоприбулих смс для керування сигналізацією;

Функція охоронної системи реалізована за допомогою датчиків газу та вогню. Якщо охоронна система спрацює (при знаходженні газу чи вогню під час включеної охоронної системи), на мобільний телефон, номер якого буде вказано в програмі, прийде текстове повідомлення з відповідним текстом та п’єзоелемент почне випромінювати звук типу сирени, світлодіод загориться.

Алгоритм роботи програми зображений на рис.3.9

Рис.3.9 — Блок-схема алгоритму роботи програми

Управління сигналізацією відбувається за допомогою смс команд. Тому після кожного циклу перевірки датчиків ми перевіряємо новоприбулі смс повідомлення, а після – видаляємо смс повідомлення з сім карти, щоб не займати її пам’ять. Після виконання циклу програми, чекаємо певної затримки потім знову повертаємося до запиту датчиків. І так безліч разів, поки сигналізація не буде вимкнута.

3.4 Розробка програмного забезпечення

В даному розділі, продемонстровано програмування *Arduino* для отримання та відправки *SMS*-повідомлень на вказаний номер телефону.

Для взаємодії з модулем ми використовували інтерфейс *UART (Serial)* за допомогою спеціальних *AT*-команд, що будуть продемонстровані нижче.

Програмний код починається з включення бібліотеки *SoftwareSerial.h* (для реалізації обміну по *UART*-інтерфейсу) та вказівці виводів до яких підключені *Tx* і *Rx* виводи модуля *SIM800L*. Підключені бібліотеки для роботи з дзвінками та смс. Оголошені змінні для підключення датчиків та зберігання значень з датчиків. Введені змінні номеру та шаблонів смс для їх швидкого редагування (рис.3.10).

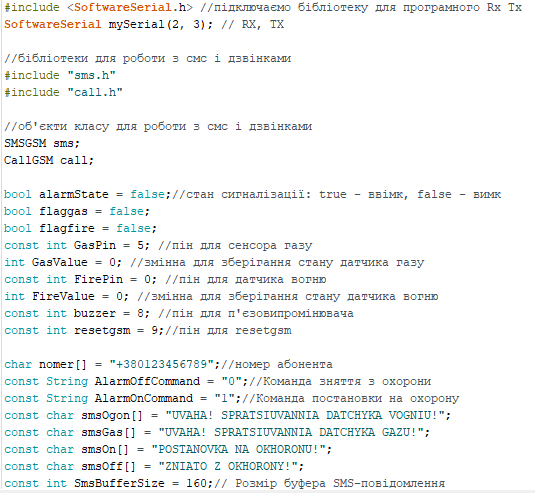


Рис.3.10 — Підключення бібліотек та оголошення змінних

Функція *void setup()* бере участь в ініціалізації системи. Тут ми вказуємо мікроконтролеру команди, які він виконає в момент завантаження (ці команди виконуються тільки один раз при старті системи).

Спочатку ініціалізуємо послідовний зв’зок з *Arduino* і *Arduino IDE (Serial Monitor),* потім ініціалізуємо послідовний зв’зок з *Arduino* та *SIM800L*. Чекаємо підключення модуля до мережі. Задаємо формат смс повідомлень та кодування тексту.

Використані *AT* - команди:

*AT* – Перевірка модуля, отримуємо у відповідь *AT*, а після *ОК*, якщо модуль відповів;

*AT + COPS?* – інформація про мобільного оператора;

*AT + CMGF = 1* - вибирає формат *SMS*-повідомлення у вигляді тексту. Формат за замовчуванням *PDU*;

*AT+CSCS=\"GSM\"* – кодування текстового режиму;

Алгоритм функції *void setup()* зображений на рис.3.11

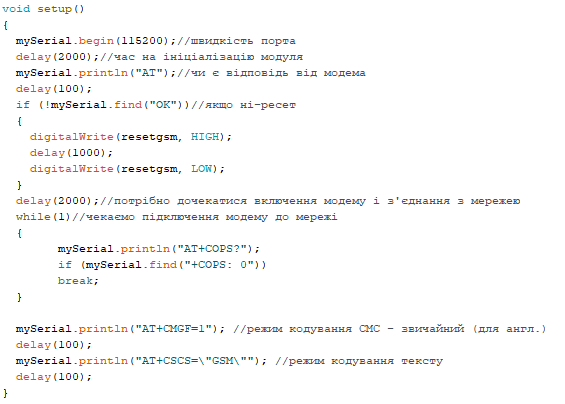
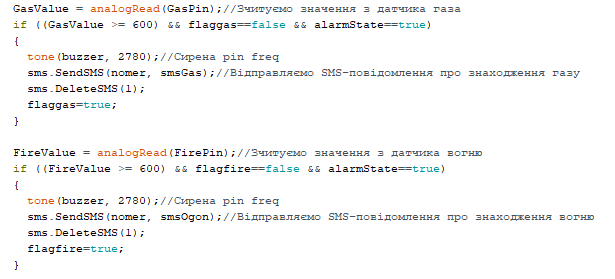


Рис.3.11 — функція *void setup()*

Функція *void loop()* повинна помістити команди, які будуть виконуватися весь час, поки включена плата *Arduino*. Почавши виконання з першої команди, мікроконтролер дійде до кінця і відразу ж перестрибне в початок, щоб повторити ту ж послідовність.

Спочатку зчитуємо значення з датчика газу. В разі спрацювання відправляємо смс користувачу та включаємо сирену. Далі перевіряємо непрочитані смс, завдяки яким ми керуємо ввімкненням/вимкненням сигналізації.

Алгоритм функції *void loop()* зображений на рис.3.12



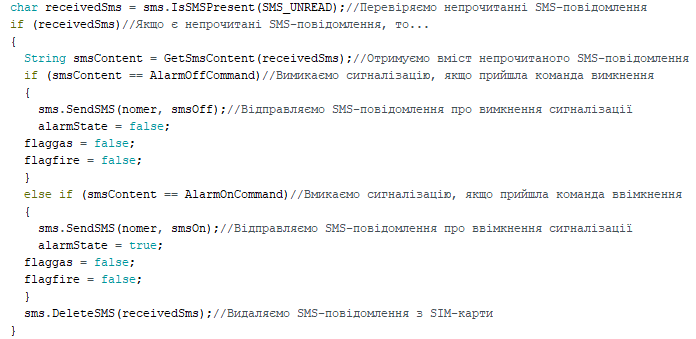


Рис.3.12 — Функція *void loop()*

Наведений лістинг демонструє основний алгоритм програми. Програма прослуховує запити, надіслані від користувачів. Після находження запиту вона виконує закладену функцію. У разі спрацювання датчика подає сигнал на *GSM* модуль, який сповіщує користувача про спрацювання. Повний лістинг програми наведений в додатку А.

3.5 Перевірка працездатності інформаційної системи протипожежної безпеки

Оскільки в кваліфікаційній роботі представлені перша версія макету та програмного забезпечення, є можливість для вдосконалення. У розділах розробки наведені рішення, які працюють, але можуть містити певні недоліки. З цього робимо висновок, що прилад можна продемонструвати, як макет. В такому разі з’являється можливість перевірки працездатності периферії та віртуальної частини, а саме взаємодія програмного коду з макетом. Також з такої моделі перевірки стає зрозуміло, що у подальшому можливо змінити або додати щось для збільшення функціоналу на базі прототипу.

Оскільки це лише прототип і тут перевіряється робота пристрою, то поки немає потреби в обмеженні габаритних розмірів. Можливо розташувати елементи так, як буде зручно (в логічній послідовності згідно блок-схеми пристрою). Для даного макету було закуплено усі необхідні модулі та з'єднання.

Для під’єднань модулів між собою використані звичайні дроти для плат *Arduino*. Більш навантажено спроможні дроти використовувати не доцільно, бо вся схема працює до 5 В.

Принципова схема підключення елементів сигналізації зображена в додатку Б.

Перевірка працездатності полягає у підключені вже готової системи сигналізації, та програмного забезпечення. Для перевірки працездатності, задовольняє умова роботи модулів без помилок та відсічок. Перевірка сигналізації здійснюється за допомогою смартфону.

Оскільки розроблений пристрій є прототипом *GSM*-сигналізації на *Arduino*, електричні з’єднання виконані за допомогою макетної плати.

Внаслідок перевірки працездатності отримано практичне підтвердження робити застосованих рішень, які в цілому задовольняють завдання до кваліфікаційної роботи.

Фінальний продукт готовий і може бути експлуатований.

Висновки за розділом 3

У даному розділі проаналізовано принцип роботи *GSM* модуля, показанопрограмний та апаратний принцип роботи пристрою, наведені основні моменти роботи приладу, розроблена блок-схема алгоритму програми, продемонстрована розробка програмного забезпечення та перевірка працездатності пристрою