

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра _____ Комп'ютерних систем та мереж _____

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних систем та
мереж

_(Олександр ЛИТВИНЕНКО)

«_____» _____ 2022 р.

ДИПЛОМНИЙ ПРОЄКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"БАКАЛАВР"**

Тема: _____ Апаратно-програмний модуль системи техногенної безпеки _____

Виконавець: _____ Дмитро ЧЕРНЕНКО _____

Керівник: _____ Віталій НЕЧИПОРУК _____

Нормоконтролер: _____ Євген ТУПОТА _____

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних систем та мереж

Напрямок (спеціальність) 123 "Комп'ютерна інженерія"

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерних систем та
мереж

_____ (Литвиненко О.Є.)

«_____» _____ 2022 р.

ЗАВДАННЯ

на виконання дипломного проєкту

Черненку Дмитру Олексійовичу

_____ (прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема проєкту (роботи): Апаратно-програмний модуль системи техногенної безпеки об'єктів інфраструктури

затверджена наказом ректора від "06" травня 2022 року № 485/ст.

2. Термін виконання проєкту (роботи): з 16.05.2022 до 19.06.2022

3. Вихідні дані до проєкту (роботи): керівництво користувача, програмний модуль, програмна документація.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1) Аналіз предметної області.

2) Структура програмно-апаратного модулю.

3) Покрокова розробка програми.

4) Висновки щодо результату.

5. Перелік обов'язкового графічного матеріалу:

Схема алгоритму роботи програмно-апаратного модулю

Структурна схема програмно-апаратного модуля

Діаграма варіантів використання програмно-апаратного модулю

Схема пристрою

6. Календарний план

<i>№ п/п</i>	<i>Етапи виконання дипломного проєкту</i>	<i>Термін виконання етапів</i>	<i>Примітка</i>
1	<i>Аналіз готових програмних рішень та альтернатив</i>	20.05.2022- 21.05.2022	
2	<i>Вибір мови програмування та середовища розробки програми</i>	22.05.2022- 23.05.2022	
3	<i>Розробка структури програми, UML діаграм та схем</i>	24.05.2022- 26.05.2022	
4	<i>Розробка апаратної частини</i>	27.05.2022- 30.05.2022	
5	<i>Тестування програмного модулю</i>	31.05.2022- 01.06.2022	
6	<i>Заповнення пояснювальної записки</i>	02.06.2022- 03.06.2022	
7	<i>Підготовка доповіді та презентації</i>	04.06.2022- 05.06.2022	

7. Дата видачі завдання _____

Керівник дипломного проєкту _____
(підпис)

Завдання прийняв до виконання _____
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту «Апаратно-програмний модуль системи техногенної безпеки об'єктів інфраструктури»: 52 сторінок, 24 рисунків, 5 таблиць, 14 використаних джерел.

*МОДУЛЬ, ДАТЧИК, ПЛАТА, БІБЛІОТЕКА, LCD ДИСПЛЕЙ,
РАДІОТЕХНОЛОГІЯ, ARDUINO, I2C*

Об'єкт – захист технічних об'єктів.

Предмет – апаратно-програмний модуль системи техногенної безпеки об'єктів інфраструктури.

Мета – розробка апаратно-програмного модулю для забезпечення захисту об'єкту.

Методи проєктування – середовище розробки *Arduino IDE*, мова програмування *Arduino*.

Результати – програмно-апаратний модуль для забезпечення безпеки об'єктів інфраструктури

ЗМІСТ

ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1_АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЄКТУВАННЯ.....	9
1.1 Типи оповіщення в існуючих системах безпеки.....	9
1.2 Існуючі технології зв'язку модулів в системах безпеки.....	10
1.3 Аналіз аналогів та готові програмні рішення в області систем безпеки.....	14
1.4 Висновки до розділу.....	17
РОЗДІЛ 2_СТРУКТУРА ПРОГРАМНО-АПАРТНОГО МОДУЛЯ	18
2.1 Обрані технології та методи створення проєкту	18
2.2 Схема алгоритму програми	28
2.3 Структура апаратного модуля	30
2.4 Висновки до розділу.....	32
РОЗДІЛ 3_РОЗРОБКА АПАРАТНОГО МОДУЛЮ	33
3.1 Покрокова розробка апаратного модулю	33
3.2 Висновки до розділу.....	45
РОЗДІЛ 4_КЕРІВНИЦТВО КОРИСТУВАЧА	46
4.1 Технічні параметри апаратного модуля “ <i>Security_system</i> ” та його тестування	46
4.2 Керівництво користувача	47
4.3 Висновки до розділу.....	49
ВИСНОВОК	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51

Кафедра КСУ				НАУ 22 28 52 000 ПЗ			
Виконав	Черненко Д.О.			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЄКТУВАННЯ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В.В.				Д	5	52
Консультант					СП-435Б 123		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Arduino – це мікроконтролер сімейства *Atmega*.

C++ – мова програмування.

SMS – служба коротких повідомлень.

LCD – рідкокристалічний дисплей.

Bluetooth – технологія бездротового зв'язку.

GSM – система подання сигналу для мобільних телефонів.

IDE – Інтегроване середовище розробки.

I2C – послідовна шина даних для зв'язку інтегральних схем.

ВСТУП

Природні екологічні і техногенні ризики були завжди. Вони мають об'єктивні причини і є наслідком розвитку еволюції. На людину і її оточення негативно впливають не тільки природні небезпеки, сама людина в процесі своєї діяльності постійно впливає на середовище, тим самим створюючи техногенні і антропогенні небезпеки. Не зважаючи на розвиток людини, в різних процесах її діяльності можливі не передбаченні обставини, які можуть призвести від полумок і не великих пошкоджень до техногенних катастроф, саме тому для уникнення подібних випадків необхідно уважно слідкувати за станом об'єктів інфраструктури системами безпеки.

Під системами безпеки прийнято розуміти комплекс пристроїв, основне призначення яких - запобігання тому щоб поломка або аварія залишилась не поміченою. Розвиток технологій значно розширює функціональність систем безпеки. Це сприяє зростанню ефективності їх роботи. Ще однією перевагою подібних систем вважається складність виведення їх з ладу. Пристрої мають підвищену міцність і стійкість до механічних пошкоджень.

Метою дипломного проєкту є створення Апаратно-програмного модулю системи техногенної безпеки на базі платформи *Arduino*. Система передбачає управління за допомогою *LCD* дисплею.

Для того аби досягти поставленої мети необхідно виконати наступні задачі:

1. Розібратися в термінології та методології, які використовуються в проєктуваннях апаратно-програмних модулів системи безпеки.
2. Вивчити літературу з наведених питань;
3. Дослідити існуючі алгоритми, проаналізувати їх роботу, виявити слабкі місця і запропонувати варіанти можливих покращень;
4. Реалізувати основні алгоритми і алгоритми з покращеннями та провести аналіз результатів.

До проєкту поставлені наступні функціональні вимоги:

1. Реалізувати режим керування: за допомогою клавіатури з *LCD* дисплеєм.

2. Реалізувати процедуру з'єднання всіх датчиків з платою.
3. Виконати повноцінне функціонування всіх складових апаратного модуля.
4. Реалізувати процедуру передачі даних з плати на *LCD* екран.
5. Реалізувати повне функціонування всіх кнопок на клавіатурі.
6. Реалізувати швидке звукове сповіщення користувача в разі підняття тривоги.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЄКТУВАННЯ

1.1 Типи оповіщення в існуючих системах безпеки.

На сьогоднішній день, існує два основних види типів сповіщення, це провідні та безпроводні охоронні системи.

Головною відмінністю перших від других, є спосіб їх з'єднання з центральною консоллю управління. У першому випадку, це відбувається через дроти, а в другому - за кошт бездротової передачі сигналів.

Провідне обладнання коштує трохи дешевше, проте воно вимагає додаткових витрат і умов по установці. Що ж стосується бездротових моделей, то вони хоч і мають трохи вищий показник ціни, зате є значно надійніше і довговічніше. До того ж, процес їх монтажу і налаштування займає всього кілька хвилин, оперативна установка не вимагає додаткових дій по проведенню проводки, свердління стін та встановлення додаткових систем.

За типом охоронні сигналізації поділяються на автономні і виведені. У першому випадку система охорони сповіщає тільки Вас за допомогою звичайних SMS-повідомлень на вказаний номер телефону. Що ж стосується виведеної системи, то вона безпосередньо з'єднується з пультом служб охорони, які при надходженні сигналу висилають оперативний загін.

Система Розумний Дім поєднує в собі два типи охоронних систем. Іншими словами, сигнал може відправлятися і на централізований пульт, і безпосередньо власнику або відповідальній особі. Периметр території, що охороняється обладнується спеціалізованими датчиками, які, в свою чергу, з'єднані з центральним блоком управління. При виникненні тієї чи іншої нештатної ситуації (спрацював датчик диму, протікання води, висока температура і т. Д.),

Кафедра КСУ				НАУ 22 28 52 000 ПЗ			
Виконав	Черненко Д.О.			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЄКТУВАННЯ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В.В.				Д	9	50
Консультант					СП-435Б 123		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

Відбувається передача сигналу від датчика на головний пульт управління. Після отримання сигналу, центральний блок відправляє автоматичне SMS-повідомлення на телефон служби охорони.

Для того, щоб поставити і зняти з охорони об'єкт, досить просто ввести правильний пароль. Налаштування системи просте і зручне, тому на повне ознайомлення з її можливостями піде не більше 10 хвилин.

1.2 Існуючі технології зв'язку модулів в системах безпеки

В системах безпеки використовують різні канали і технології передачі даних, як дротові (з використанням стандартів *HPNA*, *CEBus*, *EIB* тощо), так і канали радіозв'язку (з використанням протоколів *WAP*, *GPRS*, *EDGE* тощо).

Перспективними є системи безпеки, в яких застосовані системи на базі бездротових технологій передачі даних, основними плюсами яких є висока швидкість розгортання мережі і встановлення первинних перетворювачів, систем контролю і виконавчих механізмів, низька вартість робіт по монтажу устаткування, легка реконфігурація (бездротові первинні перетворювачі можуть встановлюватися майже в будь-якому місці в межах чутливості приймача). Як приклад системи сигналізації на базі бездротових комп'ютерних систем можна навести такі системи, що представлені на ринку України: ПЦН *GSM-900/1800* серії «Орлан» (Україна), ППКП серії «Макс» (Україна), системи фірми *Satel* (Польща) тощо.

Серед основних технологій дистанційної передачі даних виділяють: *Bluetooth*.

Bluetooth (англ. *Bluetooth*) – технологія бездротового зв'язку, розробленою групою *Bluetooth Special Interest Group (Bluetooth SIG)*, яка була заснована 20 травня 1998 році. У неї увійшли компанії *Ericsson*, *IBM*, *Intel*, *Toshiba* і *Nokia*. Потім багато компаній, включаючи *Microsoft*, *Lenovo* і *Motorola*, вступили в неї як асоційовані члени. Будь-яка компанія, яка планує розробляти пристрої *Bluetooth*, може безкоштовно увійти в цю групу. У *SIG* вже складається близько 2000 компаній. Згодом *Bluetooth SIG* і *IEEE* досягли угоди, на основі якої специфікація *Bluetooth* стало частиною стандарту *IEEE 802.15.1*. Роботи із

створення *Bluetooth* компанія *Ericsson Mobile Communication* почала в 1994 році. Спочатку ця технологія була пристосована під потреби системи *FLYWAY* у функціональному інтерфейсі між мандрівниками та системою.

Основне призначення *Bluetooth* – забезпечення економного (з точки зору споживаного струму) і дешевого радіозв'язку між різноманітними типами електронних пристроїв, таких як мобільні телефони та аксесуари до них, портативні та настільні комп'ютери, принтери та інші. Причому, велике значення приділяється компактності електронних компонентів, що дає можливість застосовувати *Bluetooth* у малогабаритних пристроях розміром з наручний годинник. *Bluetooth* забезпечує обмін інформацією між такими пристроями як кишенькові і звичайні персональні комп'ютери, мобільні телефони, ноутбуки, принтери, цифрові фотоапарати, мишки, клавіатури, джойстики, навушники, гарнітури на надійній та недорогій радіочастоті для ближнього зв'язку. *Bluetooth* дозволяє цим пристроям обмінюватись інформацією, коли вони знаходяться в радіусі від 10 до 100 метрів один від одного, навіть в різних приміщеннях. Дальність дуже сильно залежить від механічних та радіо перешкод.

Bluetooth є високошвидкісним видом зв'язку, заснованим на радіохвилях. Технологія розроблена для встановлення зв'язку між мобільними телефонами, ноутбуками та іншими портативними пристроями. На відміну від пристроїв, заснованих на технології інфрачервоного випромінювання, *Bluetooth* не вимагає знаходження з'єднуються пристроїв на лінії проходження променя.



Рисунок 1.1 – принцип роботи технології *bluetooth*

Технологія *Bluetooth*, в якому сенсі, є модифікованою різновидом існуючої технології локальних мереж. З іншого боку, *Bluetooth* зручніше в силу невеликих розмірів і невисокої вартості.

В даний час чіп *Bluetooth* являє собою мікросхему розміром в 0,9 квадратних сантиметрів. Ведуться розробки нового, ще більш мініатюрного чіпа, виробництво якого почнеться незабаром. Очікується, що ціни на пристрої *Bluetooth* помітно знизяться. Чіпи *Bluetooth* зараз вбудовуються в багато пристроїв. Недорогі мініатюрні передавачі розміщені в цифрових пристроях. Пристрої, що використовують технологію *Bluetooth*, працюють на частоті 2,45 Гц при здійсненні з'єднання.

Технологія *Bluetooth* підтримує досить високу швидкість передачі даних і кілька голосових каналів. Чіп *Bluetooth* або безпосередньо вбудовується в пристрій, або використовується в якості адаптера. У випадку з комп'ютером, його можна використовувати з портом *USB*. Кожен пристрій *USB* має 48-бітову адресу зі стандартів *IEEE 802*. З'єднання можна встановлювати як з одним

пристроєм, так і з кількома пристроями одночасно. Стандартний діапазон роботи *Bluetooth* не перевищує 10 метрів, але якщо збільшити потужність, діапазон роботи може збільшитися до 100 метрів.

Принципи роботи систем радіоуправління.

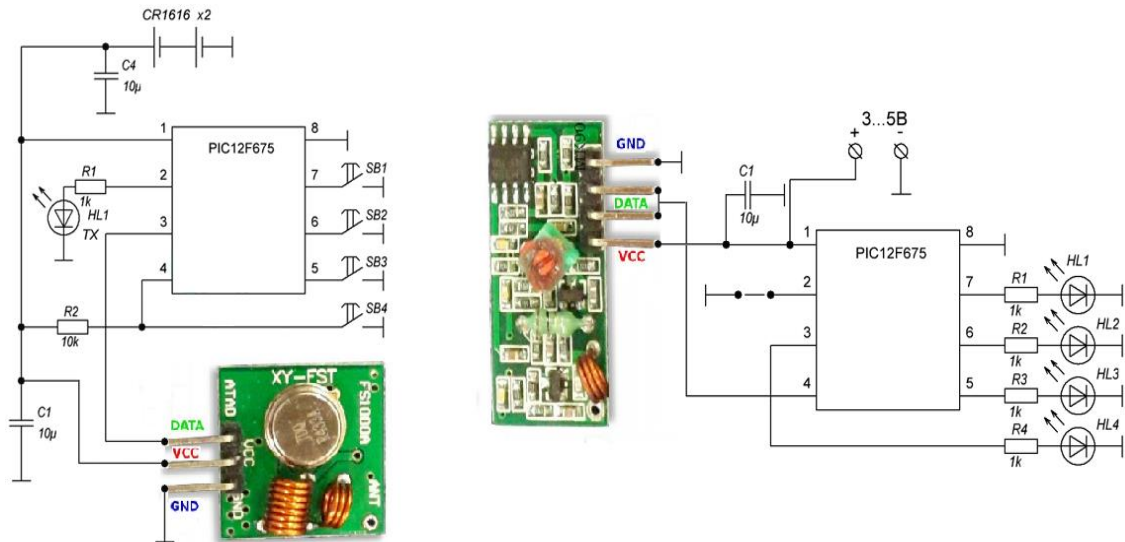


Рисунок 1.2 – Пристрій радіоуправління на 4 команди

У системах радіоуправління при передачі команди від оператора, (диспетчера) до об'єкта, команди, що задаються оператором деякої маніпуляцією на пульті управління, перетворюється в послідовність електричних імпульсів, а потім цими імпульсами модулюється (за допомогою фазової, амплітудної, частотної модуляції і т. Д.) несуча частота радіосигналу.

Для підвищення надійності радіоуправління застосовують різні перешкодостійкі коди, в тому числі коригувальні коди, а також контроль факту прийому або виконання надісланих команди через зворотний радіоканал, при цьому від об'єкта управління до пункту управління передаються сигнали, що підтверджують прийом і виконання (або тільки прийом, або тільки виконання) переданої команди (так зване квотування).

Системи радіоуправління поділяються на дискретні (передають фіксований набір команд на включення / вимикання виконавчих пристроїв, або зміна їх стану на заданий крок) і пропорційні (передають команди на плавно змінюваний відгук виконавчого пристрою, наприклад рульового механізму, відповідно до зміни положення органу управління).

У деяких системах (наприклад, в системах управління ракет в польоті, безпілотних літальних апаратів) управління здійснюється безперервно за допомогою автоматично одержуваного сигналу неузгодженості між заданим і дійсним (поточним) станами об'єкта управління, наприклад, положення його в просторі.

Проте при розробці та побудові бездротових систем сигналізацій на базі нових технологій бездротової передачі даних виникає чимало важливих науково-технічних питань, які потребують вирішення: оптимальне розміщення первинних перетворювачів в мережі, вибір типу бездротової технології передачі даних, забезпечення завадостійкості, гарантований зв'язок між передавачами мережі для надійної передачі інформації, довготривала автономна робота компонентів системи тощо. Отже розробка нових методів моделей та компонентів комп'ютерних систем для підвищення ефективності роботи системи безпеки на базі бездротових технологій є вельми актуальною.

1.3 Аналіз аналогів та готові програмні рішення в області систем безпеки

Так як в кожній компанії яка володіє якоюсь інфраструктурою є своя система безпеки, і очевидно, яка саме в більшості випадків не розголошується за для більшої безпеки, мною було оглянуто системи сигналізацій в “Розумному домі”, так як принцип цих двох систем майже не відрізняється, і такі аналоги підходять під тему.

На даний момент у світі є багато готових рішень. При виборі варто уважно ознайомитися з функціями, вивчити відгуки на форумах. Система безпеки не найдешевше задоволення.

Ajax. Компанія *Ajax Systems* одна з найвідоміших компаній, як в Україні так і в світі, по виготовленню бездротових систем безпеки. Продукт компанії — комплексна професійна система безпеки *Ajax*.



Рисунок 1.3 – Бездротова сигналізація *AJAX systems*

Модельний ряд *Ajax* 2018 року складається з 21 пристрою, які забезпечують захист від пограбування, пожежі та затоплення, а також допомагають керувати електроживленням. Для зв'язку пристрої системи використовують розроблений *AjaxSystems* пропрієтарний радіопротокол *Jeweller* з унікальною для охоронної сфери сукупністю характеристик: двосторонній зв'язок на відстані до 2000 метрів, термін роботи датчиків від батарей до 5-7 років, передача тривоги за 0,15 секунди, легко може бути розширений додатковими датчиками екосистеми *Ajax* для вирішення широкого кола охоронних завдань. Два активних каналу - *GSM* і *Ethernet* | повідомляє про тривоги за допомогою *push*-повідомлень, *SMS* або телефонного дзвінка, до 50 користувачів системи, що особливо зручно для використання в офісах, захищений від глушіння, а дані, що передаються зашифровані.

LifeSmart Security. Комплект для побудови високоякісної системи безпеки розумного будинку або офісного приміщення.

Не настільки популярна і відома система безпеки як *Ajax* але набагато дешевша. Звісно чим популярніша і просунутіша система безпеки, тим вищий рівень безпеки об'єкту, але часто такі гіганти в своїй сфері як *Ajax* не по кишені багатьом людям, тому на допомогу приходить більш бюджетний варіант системи безпеки.

8 FEATURES

of Smart Home Security System



Рисунок 1.4 – *LifeSmart Security*

Бездротовий зв'язок виконується на частоті 433 МГц на відстані до 400 метрів в умовах відкритої місцевості. Напруга на вході 5 Вольт. Сила струму 2 Ампера. Потужність споживання всього 1,5 Ватта.

Датчик руху *LifeSmart Cube Motion Sensor LS062WH* являє собою високоточний пристрій постійно реагують на рух всередині будинку або офісного приміщення. Підключивши такий прилад до системи розумного будинку *LifeSmart* він перетворюється в потужний інструмент, який дає старт включенню освітлення або відеокамери, а також виконання різних ситуацій при визначенні руху в приміщенні. Автономність роботи покликаний підтримувати *Li-Ion* акумулятор з ємністю 850mAh. Бездротовий зв'язок досягає максимум в відстані 400 метрів в умовах відкритої місцевості.

Датчик відкриття чи інакше іменований магнітогерконовий датчик *LifeSmart Cube Door / Window Sensor LS058WH* являє собою високоточний пристрій реагують на відкриття або закриття дверей або вікна всередині будинку або офісного приміщення. Підключивши такий прилад до системи розумного будинку *LifeSmart* він перетворюється в потужний інструмент, який дає старт включенню освітлення або відеокамери в залежності від відкриття або закриття

дверей. Автономність роботи покликаний підтримувати *Li-Ion* акумулятор з ємністю 850mAh. Бездротовий зв'язок досягає максимум в відстані 400 метрів в умовах відкритої місцевості.

Як можемо бачити, комплектуюча майже не відрізняється від інших високорівневих систем безпеки, але в *LifeSmart* не такий красивий і зручний дисплей як в наприклад *Ajax*.

1.4 Висновки до розділу

1. Проаналізовано сучасні програмні рішення в області проектування систем безпеки. Отримано основні відомості про раціональне, спрощене та продуктивне створення програмно-апаратних модулів з використання платформи *Arduino*.

2. Проаналізовано технології створення апаратних модулів керування технічним об'єктом. Для реалізації даного проекту вибрано редіотехнологічний спосіб передачі даних.

3. Відповідно до сучасних вимог з метою задоволення попиту на програмне забезпечення такого типу сформульоване завдання проектування метою якого є розробка апаратно-програмного та додержання усіх технологічних етапів розробки.

РОЗДІЛ 2

СТРУКТУРА ПРОГРАМНО-АПАРТНОГО МОДУЛЯ

2.1 Обрані технології та методи створення проєкту

Апаратна частина створена мовою програмування *Arduino* на базі платформи *Arduino UNO*. Також для створення прототипу необхідні, *GSM*-модуль *SIM800L*, зумер, *I2C* модуль, клавіатура, *LCD* дисплей, датчик температури і вологості *DHT11*, датчик газу *MQ-2*, акумулятор 5V та акумулятор 3.7V.

Для реалізації програмної частини апаратного модуля обрано мову програмування *Arduino* в середовищі розробки – *Arduino IDE*.

Синтаксис і логіка мови програмування між звичайним середовищем розробки *Arduino IDE* і середовищем розробки в *Tinkercad* майже не відрізняється, але деякі відмінності все таки існують. Наприклад деякі бібліотеки які або необхідні, або зручніші в використанні ніж ті що є в *Tinkercad* не можливо встановити, тому що використовувати дозволяється тільки вбудовані бібліотеки, в свою чергу більша частина існуючих модулів, датчиків та інших пристроїв не існує в інструментарії.

Кафедра КСУ				НАУ 22 28 52 000 ПЗ			
Виконав	Черненко Д.О.			СТРУКТУРА ПРОГРАМНО- АПАРТНОГО МОДУЛЯ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В.В.				Д	18	52
Консультант					СП-435Б 123		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						



Рисунок 2.1 — Плата *Arduino UNO*

Платформа *Arduino* та її мова програмування.

Мова програмування *Arduino* є стандартним *C ++* (використовується компілятор *AVR-GCC*) з деякими особливостями, які полегшують новачкам написання першої працюючої програми.

Програми, написані програмістом *Arduino*, називаються начерки (або скетчі — від англ. *Sketch*) і зберігаються в файлах з розширенням *ino*. Ці файли перед компіляцією обробляються препроцесором Ардуіно. Також існує можливість створювати і підключати до проєкту стандартні файли *C ++*.

Обов'язкову в *C ++* функцію *main ()* препроцесор *Arduino* створює сам, вставляючи туди необхідні «чорнові» дії.

Програміст повинен написати дві обов'язкові для *Arduino* функції *setup ()* і *loop ()*. Перша викликається одноразово при старті, друга виконується в нескінченному циклі.

В текст своєї програми (скетчу) програміст не зобов'язаний вставляти заголовки використовуваних стандартних бібліотек. Ці заголовки додасть препроцесор *Arduino* відповідно до конфігурації проєкту. Однак призначені для користувача бібліотеки потрібно вказувати.

Менеджер проекту *Arduino IDE* має нестандартний механізм додавання бібліотек. Бібліотеки у вигляді вихідних текстів на стандартному *C ++* додаються в спеціальну папку в робочому каталозі *IDE*. При цьому назва бібліотеки додано до списку бібліотек в меню *IDE*. Програміст зазначає потрібні бібліотеки і вони вносяться до списку компіляції.

Arduino IDE не пропонує ніяких налаштувань компілятора і мінімізує інші налаштування, що спрощує початок роботи для новачків і зменшує ризик виникнення проблем.

Найпростіша *Arduino*-програма складається з двох функцій:

setup (): функція викликається одноразово при старті мікроконтролера.

loop (): функція викликається після *setup ()* в нескінченному циклі весь час роботи мікроконтролера.

В якості центрального контролера був обраний *Arduino*, так як це недорогий мікропроцесор з відкритою архітектурою, здатний зчитувати вхідні дані у вигляді напруги на своїх аналогових контактах. Якщо до певних входів пристрою підключити датчики, то воно програмним способом зчитувати інформацію з цих контактів. Огляд *Arduino* призначене для створення електронних пристроїв, що працюють за заданим алгоритмом з можливістю реагувати на зовнішні сигнали.

Arduino – це платформа з відкритою принциповою схемою, яка доступна як для підключення інших пристроїв, так і для зміни самої електричної схеми пристрою.

Таблиця 2.1 – Загальні технічні характеристики

Напруга живлення	Від 5 до 12В
Максимально допустимий струм портів	20мА
Максимально допустима напруга на портах	від -0.5В+5.5В
Струм спрацьовування запобіжника <i>USB</i> входу	500А
Керуючий мікроконтролер	<i>Atmega 328</i>
Цифрові порти вводу/виводу	14 портів

Аналогові порти вводу	6 портів
ППЗУ(<i>Flash Memory</i>)	32 Кб
ОЗУ	2Кб
ПЗУ	1Кб
Тактова частота	16 МГц

GSM-модуль SIM800L

Плата на базі *GSM / GPRS* модуля *SIM800L* - простий засіб для підключення різних пристроїв до *GSM* мережі.

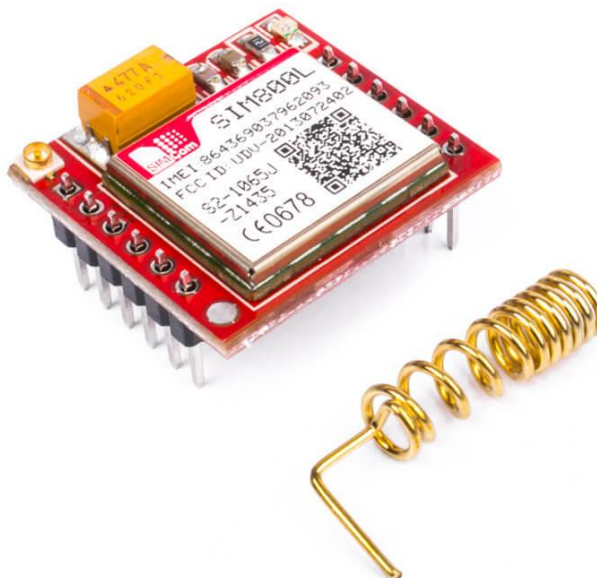


Рисунок 2.2 — *GSM* модуль *SIM800L*

Плата має всю необхідну обв'язку, а також вбудований роз'єм для карти *micro-SIM*. За допомогою даного модуля можна приймати і відправляти *SMS*, здійснювати дзвінки, відправляти дані через *E-Mail* і т.д.

Таблиця 2.2 – Характеристики модулю

Напруга живлення	Від 3.7 до 4.2 В
Підтримувані діапазони	<i>GSM</i> 850/900/1800/1900 МГц
Клас потужності	4 (2 Вт в діапазонах 850/900 МГц)
Розміри	23 * 25 * 7 мм

Датчик температури і вологості *DHT22*

Ці сенсори прості і повільні, але при цьому відмінно підходять для хобі-проектів на *Arduino*. Датчики *DHT* складаються з двох основних частин: емнісний датчик вологості і термістор. Також в корпусі встановлений простенький чіп для перетворення аналогового сигналу в цифровий.

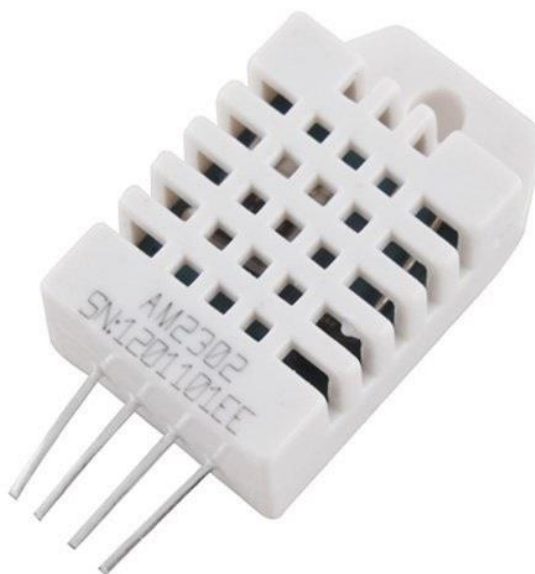


Рисунок 2.3 — Датчик температури і вологості *DHT22*

Зчитувати цифровий сигнал на виході досить просто, можна використовувати будь-який контролер, не обов'язково *Arduino*.

Таблиця 2.3 – Характеристики модулю

Напруга живлення	Від 3 до 5 В
Максимально споживаний струм	-2.5мА
Розрахований на вимірювання рівня вологості в діапазоні	Від 20% до 80%
Розміри	15 * 12 * 5.5 мм
Вимірює температуру в діапазоні	Від 0 до 50

Датчик витоку газу *MQ2*

Датчик *MQ-2* *Arduino* дозволяє виявити мінімальну концентрацію водню

і вуглеводневих газів (пропану, метану, бутану) в повітрі. Датчики *MQ-2* використовуються в проєктах розумного будинку для своєчасного виявлення газу або диму. Датчик відноситься до сімейства датчиків *MQ*, які характеризуються низькою вартістю, простотою використання і простотою підключення до мікроконтролера *Arduino*.



Рисунок 2.4 — Датчик витоку газу *MQ2*

При вимірюванні газів термін «концентрація» використовується для опису кількості газу в повітрі за об'ємом. Найбільш поширеними одиницями вимірювання є частка на мільйон і процентна концентрація. Частка на мільйон (*ppm*) - це відношення одного газу до іншого. Наприклад, концентрація 1000 *ppm* CO означає, що на кожні 999 000 молекул газу 1000 з них будуть вуглекислим газом.

Таблиця 2.4 – Характеристики модулю

Напруга живлення	5 В
Чутливість	300-10000 ppm
Робоча температура	Від: -10 до +50 °C
Вологість повітря	Не більше 95%

Максимально споживаний струм	180мА
------------------------------	-------

Зуммер АМР- В008

Вібраційний невеликої потужності, перетворювач постійного струму у змінний, який завдяки вібрації контакту переривача видає своєрідне дзижчання.

У зуммера є дуже багато переваг, які можна використовувати в різних складних ситуаціях.

1. Оскільки немає рухомої контактної частини, тому довгий термін служби, висока надійність, безперервне використання до 10 000 годин, є майже вічним пристроєм.
2. Не створює радіочастотний шум, не заважає іншій лінії.
3. Не приведе до ослаблення великої вібрації.
4. З електронним управлінням ланцюга, тому може випромінювати різноманітність приємного звуку і аналогового звуку, переривчастий звук. Чистий звук, нелегко покритий шумом.
5. Малий обсяг. Обсяг може досягати 70 дБ / 20 см, а товщина компонента тільки на 1 мм нижче.
6. Широкий діапазон робочих температур.
7. Прості в установці, електромагнітна котушка і рухається кільцем, не потрібно хвилюватися про погіршення ізоляції і відсутність витoku можливості.



Рисунок 2.5 – Найпростіший зуммер

Видає найширший набір різного роду звуків, включаючи навіть дзижчання і пицання. Прилад дуже простий при постійному використанні, особливих зусиль до його роботі докладати не доведеться, тобто не потрібно паяти або займатися макетною платою, вже все зроблено спочатку.

Символьний *LCD* дисплей *HD44780*

Рідкокристалічний дисплей — це електронний пристрій візуального відображення інформації (дисплей), принцип дії якого ґрунтується на явищі електричного переходу Фредеріка в рідких кристалах. Дисплей складається з довільної кількості кольорових або монохромних точок (пікселів), і джерела світла або відбивача (рефлектора).

Кожна з кольорових точок рідкокристалічного дисплея складається з кількох комірок (як правило, з трьох), попереду яких встановлюються світлові фільтри (найчастіше — червоний, синій і зелений). Тобто колір певної точки і її яскравість визначається інтенсивністю світіння комірок, з яких вона складається.

Дисплей виконаний на чипі, що відповідає стандартам *HD44780*, який є де-факто стандартом для *LCD*-екранів.

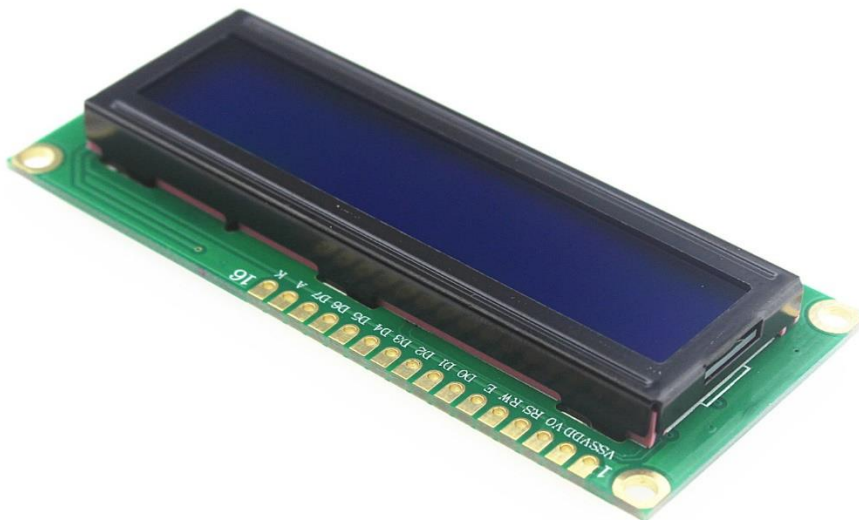


Рисунок 2.6 – Символьний *LCD* дисплей

Таблиця 2.5 – Характеристики модулю

Напруга живлення	5 В
Максимально споживаний струм підсвітки	100мА
Розмір символу	4.35 * 2.95 мм
Формат	16 x 2
Розміри	80*36 мм
Інтерфейс	HD44780
Робоча температура	0 – 50°C

Розмір точки	0.5 * 0.5 мм
Максимально споживаний струм	1мА

Матрична клавіатура

У світі електроніки клавіатурою називають пристрій, що складається з окремих кнопок, яке дозволяє вводити інформацію в комп'ютер або в інший електронний прилад.

Клавіатури мають масу варіантів виконання. На старих персональних комп'ютерах використовувалися клавіатури з механічними клавішами, які мали великий хід і були майже так само зручні як клавіші на старих друкарських машинках. Останні 15-20 років широко застосовуються мембранні клавіатури, які простіше і дешевше у виготовленні. Саме така клавіатура використовується в модулі.

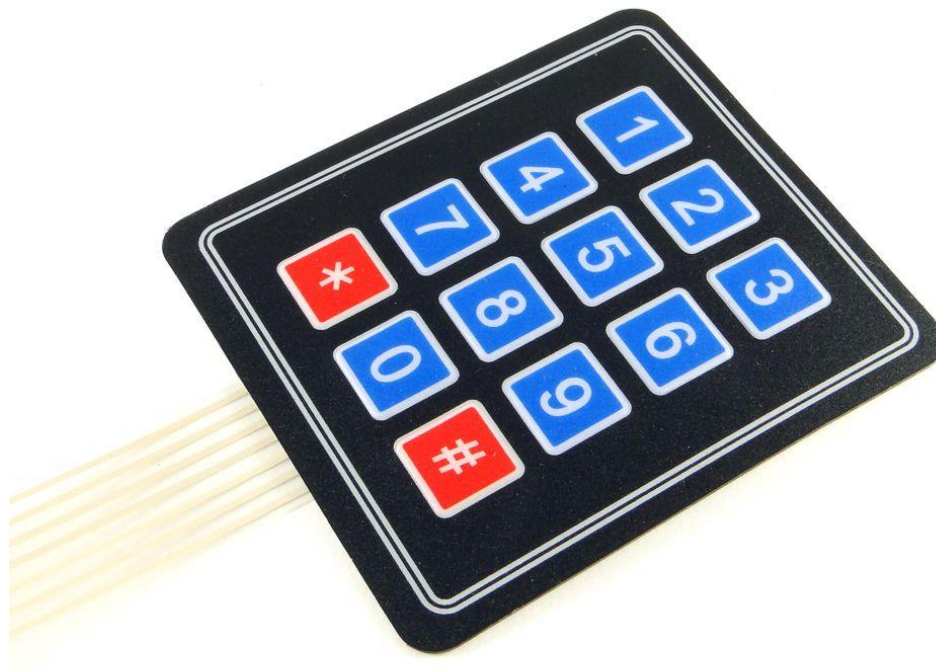


Рисунок 2.7 – Матрична клавіатура

Популярна в Ардуіно проєктах матрична мембранна клавіатура має 12 або 16 кнопок. З нижньої частини клавіатури виходить шлейф, який закінчується *dupont* розеткою.

Матрична клавіатура складається з кнопок, що утворюють матрицю $m \times n$, тобто таблицю, де m - кількість рядків, а n - кількість стовпців.

2.2 Схеми алгоритмів програми

Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми.

Для комп'ютерних програм алгоритм є списком деталізованих інструкцій, що реалізують процес обчислення, який, починаючи з початкового стану, відбувається через послідовність логічних станів, яка завершується кінцевим станом. Перехід з попереднього до наступного стану не обов'язково детермінований – деякі алгоритми можуть містити елементи випадковості. Розроблено та показано схему алгоритму програми. Блок – схема алгоритму – це графічне представлення логічної структури алгоритму, де кожний етап обробки інформації зображується у вигляді геометричних символів (блоків).

Існують правила зображення блок-схем алгоритмів. Кожен алгоритм має початок та кінець. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

Схема алгоритму показана на рисунку 2.8

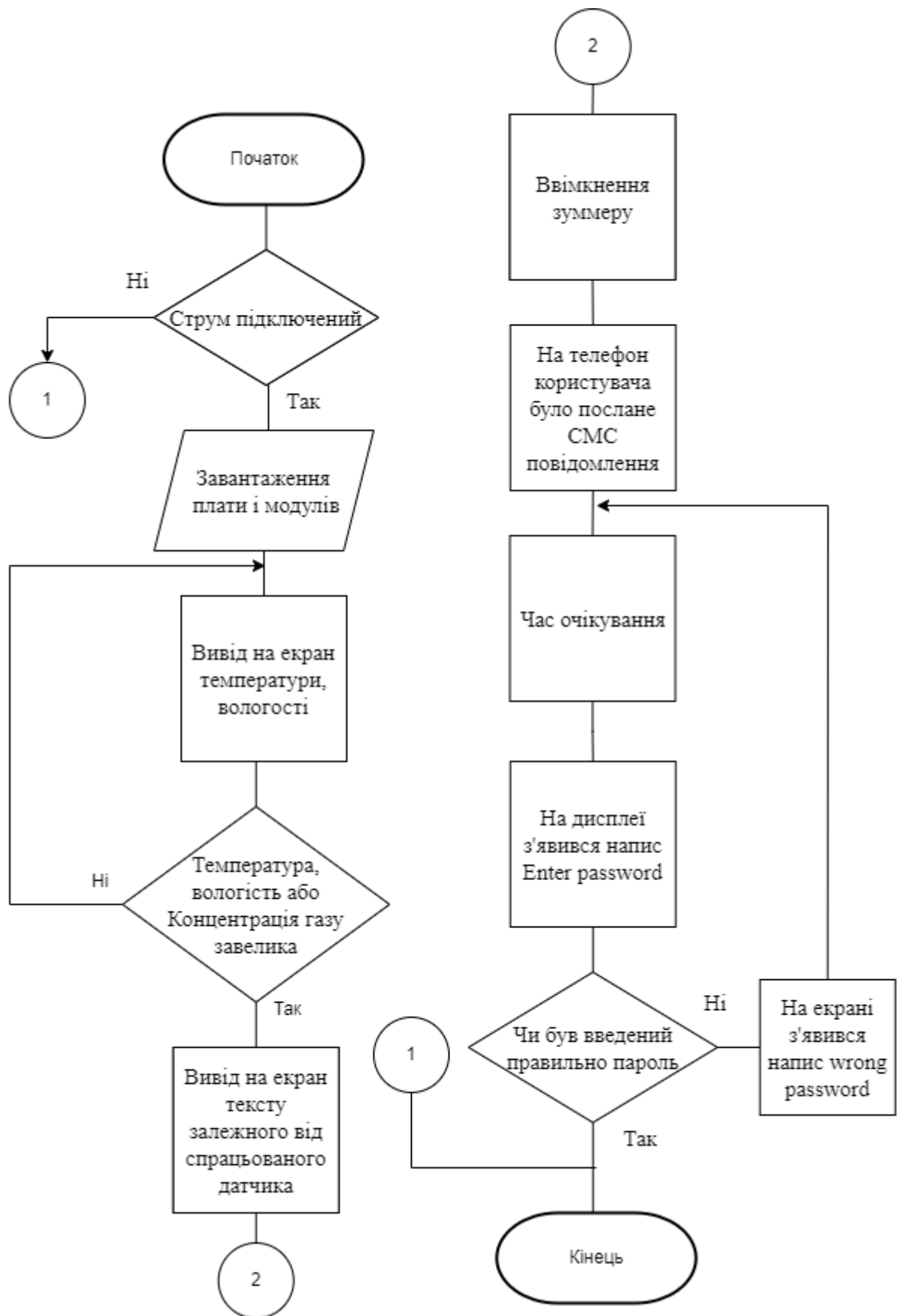


Рисунок 2.8 – Схема алгоритму

2.3 Структура апаратного модуля

Провівши аналіз готових рішень, вирішено використати наступні ресурси для створення прототипу: апаратна частина створена мовою програмування *Arduino* на базі платформи *Arduino UNO*.

Також для створення прототипу необхідні: *GSM*-модуль *SIM800L*, зумер, *I2C* модуль, клавіатура, *LCD* дисплей, датчик температури і вологості *DHT22*, датчик газу\диму та блок живлення 3.7V.

Структурна схема – схема, яка визначає основні функціональні частини виробу, їх взаємозв'язки та призначення. Під функціональною частиною розуміють складову частину схеми: елемент, пристрій, функціональну групу, функціональну ланку.

Структурна схема призначена для відображення загальної структури пристрою, тобто його основних блоків, вузлів, частин та головних зв'язків між ними. Із структурної схеми повинно бути зрозуміло, навіщо потрібний даний пристрій і як він працює в основних режимах роботи, як взаємодіють його частини.

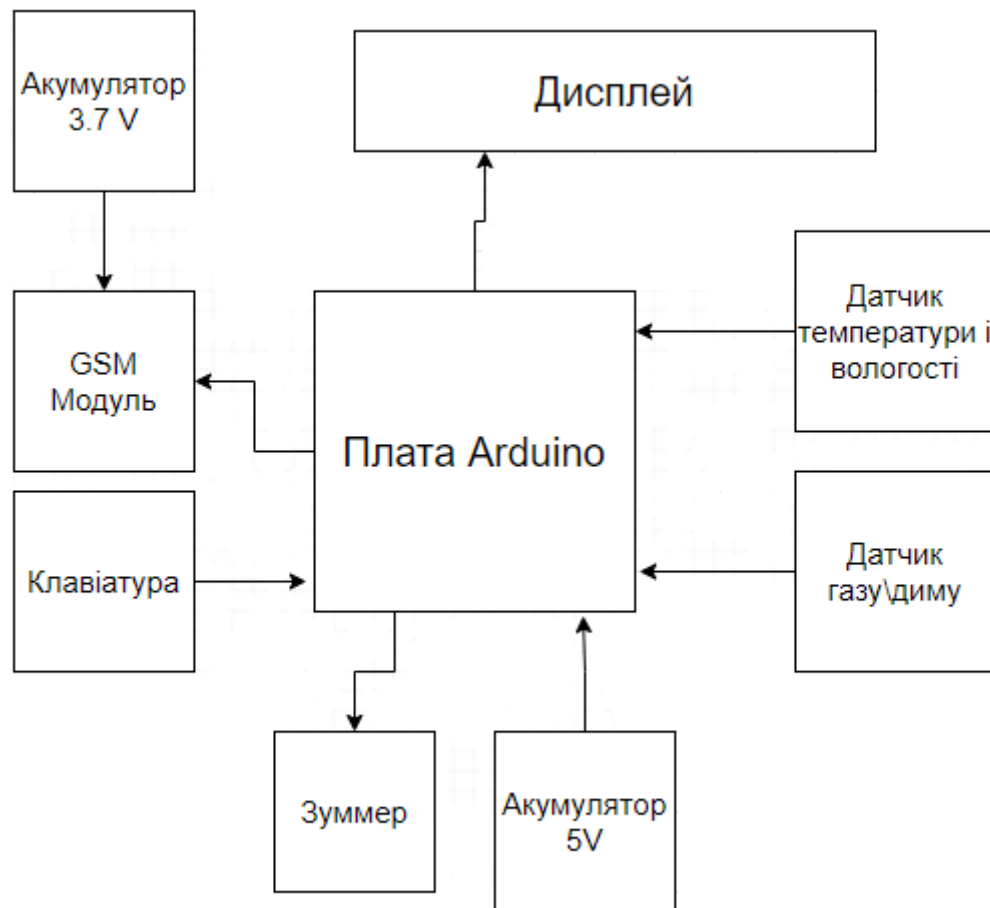


Рисунок 2.9 — Структурна схема програмно-апаратного модуля

Діаграма варіантів використання.

Варіант використання описує, з точки зору чинної особи, групу дій в системі, які призводять до конкретного результату. Варіанти використання є описами типових взаємодій між користувачами системи і самою системою. Вони відображають зовнішній інтерфейс системи і вказують форму того, що система повинна зробити.

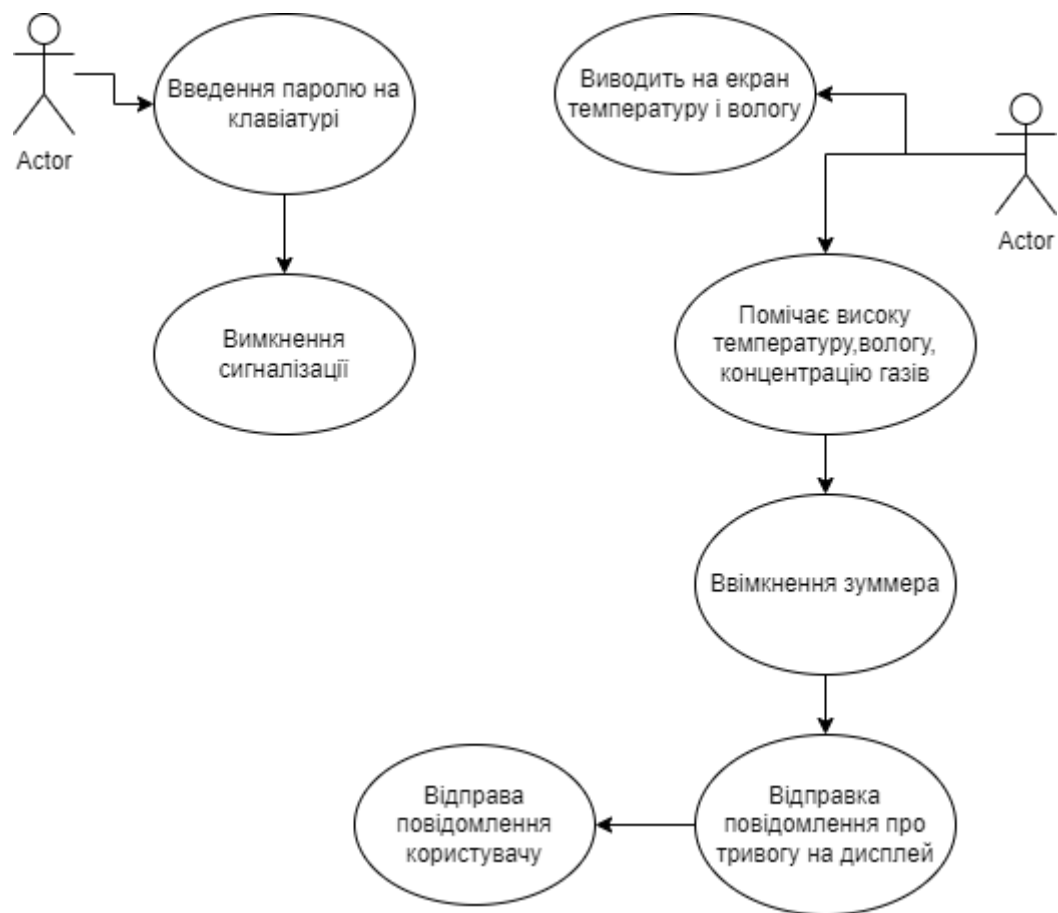


Рисунок 2.10 — Діаграма варіантів використання

2.4 Висновки до розділу

1. Описано функціональні вимоги апаратного модуля, обрані технології та методи створення проекту.

2. Розроблено структуру апаратно-програмного модулю техногенної безпеки, в якій містяться деталі та компоненти програмного модулю. Пристрій має апаратну та програмну складові.

3. Проведено підбір та аналіз характеристик всіх необхідних модулів для апаратної частини.

4. Наведено діаграми варіантів використання програмного модуля.

5. Наведено схему алгоритму програмного модуля.

РОЗДІЛ 3

РОЗРОБКА АПАРАТНОГО МОДУЛЮ

3.1 Покрокова розробка апаратного модулю

На першому етапі проектування проведено збір усіх деталей, необхідних для виконання поставленого завдання(рис3.2).



Рисунок 3.1 – Необхідні деталі

Перелік необхідних деталей:

Кафедра КСУ				НАУ 22 28 52 000 ПЗ			
Виконав	Черненко Д.О.			РОЗРОБКА АПАРАТНОГО МОДУЛЮ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В.В.				Д	33	52
Консультант					СП-435Б 123		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

1. Датчик температури і вологості *DHT11*.
2. *GSM*-модуль *SIM800L*.
3. *I2C* модуль.
4. Акумулятор 3.7V.
5. Плата *Arduino UNO*.
6. *LCD* дисплей.
7. Клавіатура.
8. Зумер.
9. Датчик газу *MQ-4* (бутан, пропан, метан, дим).
10. Блок живлення.

Наступний етап передбачав проектування схеми підключення всіх складових пристрою в єдине ціле. Далі на рис 3.2 наведена схема апаратного модуля.

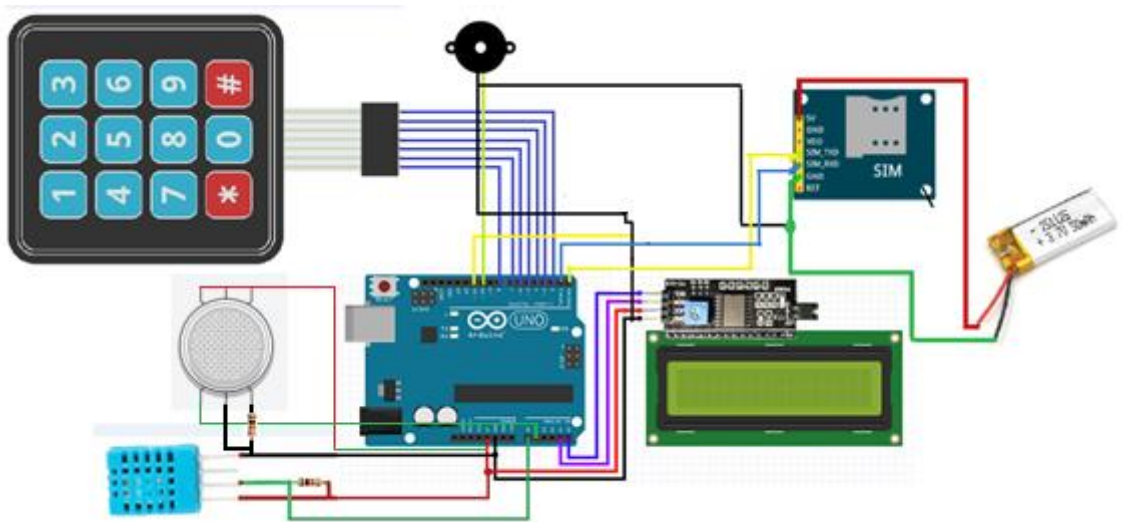


Рисунок 3.2 – Схема пристрою

В процесі програмування апаратної частини використовується, вбудоване в веб-середовище, для програмування платформи *Arduino - Arduino IDE*.

Для того щоб датчик температури і вологи працював правильно потрібно в середовищі розробки *Arduino IDE* встановити бібліотеку *DHT.h*.

Для встановлення бібліотеки потрібно перейти на вкладку “Скетч”, потім у випадаючому списку вибрати “Підключити бібліотеку” і зразу ж перейти на

“Керування бібліотеками”, далі в полі пошуку шукаємо необхідну нам *DHT.h*.

Таким чином потрібно завантажити також бібліотеки для *LCD* дисплею, *GSM* модулю, клавіатури та *I2C* модулю.

Завантаживши необхідні бібліотеки необхідно підключити їх в код програми:

```
#include <SoftwareSerial.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
```

Далі потрібно вказати адрес *LCD* дисплею і тип екрану:

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Створюємо змінну, в дужках вказуючи номери цифрових портів, по яких відбувається передача інформації:

```
SoftwareSerial Sim800(2,3); // 2 — RX Arduino (TX SIM800L), 3 — TX
Arduino(RX SIM800L)
```

Це означає, що на другий цифровий пін - *D2* підключимо пін *TX SIM800L*, а на *D3* – *RX SIM800L*.

В *LCD* дисплей запрограмовано багато різних символів, але якщо потрібен символ який не запрограмований в дисплей то є можливість його створити:

```
byte degree[8] =
{
  B00110,
  B01001,
  B01001,
  B00110,
  B00000,
  B00000,
  B00000,
};
```

За допомогою цього коду створено знак градуса для дисплею який буде стояти в кінці чисельного значення.

Далі потрібно підключити кожен кнопочок клавіатури до цифрових пінів:

```
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'#','0','*'},
  {'9','8','7'},
  {'6','5','4'},
  {'3','2','1'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {8, 7, 6};
```

Далі йде багато змінних з різними типами даних які застосовуються в логічних конструкціях і циклах:

```
int cou = 0;
int helpa = 0;
bool alarm = false;
bool pass10 = false;
bool pass20 = false;
bool pass30 = false;
bool pass40 = false;
char sirena[] = "Sirena vkluchena!";
char fsirena[] = "Temperatura >50!";

char pos;
char sendsms[160];
int dVigPin = 11;
int piiPin = 13;
int val = 0;
```

```
char pass1 = '1', pass2 = '2', pass3 = '3', pass4 = '4';
```

Тепер починається одна з основних функцій *void setup()*. Код вище обробляється платою ще до початку роботи програми, а після функції *void setup()* код буде оброблятися в момент запуску програми.

Далі підключаємо в роботу *Serial.monitor*, за допомогою якого відбувається налаштування *gsm* модулю:

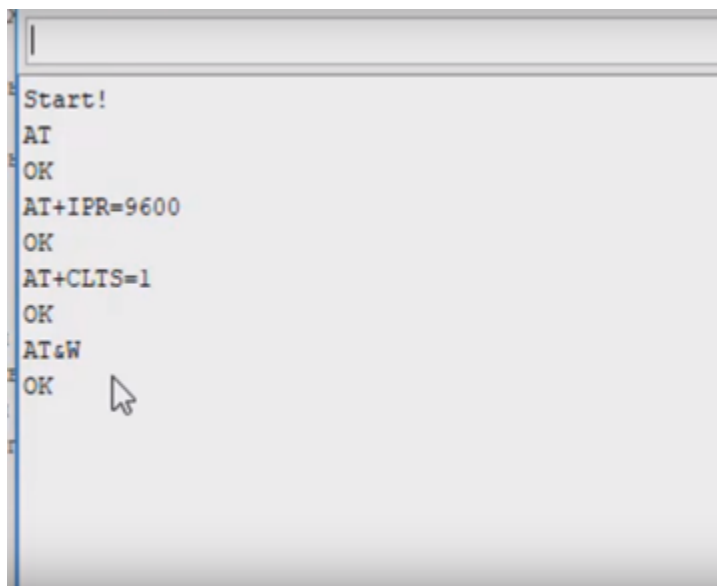
```
Serial.begin(9600);
```

9600 – швидкість обміну даними з комп'ютером.

Налаштовуємо швидкість передачі даних *gsm* модуля з комп'ютером;

```
Sim800.begin(9600);
```

Опісля проводимо перевірку працездатності модулів з допомогою AT-команд та послідовного монітора порта.



```
Start!  
AT  
OK  
AT+IPR=9600  
OK  
AT+CLTS=1  
OK  
AT&W  
OK
```

Рисунок 3.3 – Відправка тестових команд *gsm*-модулю

Далі використовуємо функцію *Serial.available()*, яка отримує кількість байт (символів) доступних для читання з послідовного інтерфейсу зв'язку. Це ті байти які вже надійшли і записані в буфер послідовного порту.

Для зберігання цих байтів створюємо перемінну, якій будуть присвоєні всі байти, які зчитуються з серійного монітору та робимо перевірку на кількість отриманих байтів:

```
if (Serial.available() > 0) {
```

```
int incomingByte = Serial.read();
```

```
}
```

Далі переходимо до отримання команд безпосередньо через модуль *gsm*:

```
boolean isStringMessage = false; // змінна для обробки SMS
```

Робимо перевірку на отримання даних від модуля *SIM800L*:

```
if (!Sim800.available())// якщо немає даних від модуля SIM800L, то далі не
```

йдемо.

```
return;
```

Всі отримані символи записуємо в нову змінну *Symb*:

```
char Symb = Sim800.read();
```

Далі проходить перевірка на завершення отримання повідомлення

```
if( '\r' == Symb) // якщо отримано символ переводу в початок нового рядка,
```

це означає, що отримання завершено

```
{
```

```
if (isStringMessage) { // якщо поточний рядок – повідомлення, то
```

відбувається наступна дія

```
}
```

Налаштовуємо піни датчика газу і пищалки на вхід *INPUT* і вихід *OUTPUT*:

```
pinMode(dvigPin, INPUT);
```

```
pinMode(piiiPin, OUTPUT);
```

Далі програмування дисплею:

```
lcd.begin();
```

```
lcd.createChar(1, degree);
```

```
dht.begin();
```

Основні команди для ввімкнення дисплею через модуль *I2C* а також *lcd.createChar(1, degree);* для застосування символу температури який був створений раніше.

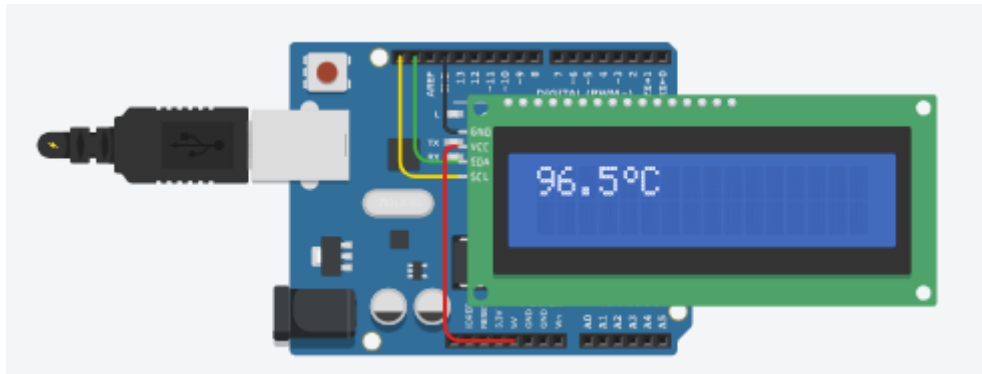


Рисунок 3.4 – Вигляд створеного знаку градусу

Тепер починається друга головна функція *void loop()*, яка являється постійним безперервним циклом.

Оголошуємо змінну типу *char* для розпізнавання клавіатури:

```
char key = keypad.getKey();
```

Тепер створюємо змінні для отримання даних з датчику температури:

```
float h = dht.readHumidity();
```

```
float t = dht.readTemperature();
```

Перша змінна відповідає за вологість а друга за температуру по цельсію.

Задаємо значення змінній яка відповідає за датчик газу:

```
val = digitalRead(dvigPin);
```

Далі йде інструкція яка при перегріві датчику викликає тривогу:

```
if(t >= 50)
```

```
{
```

```
    alarm = true;
```

```
    sms.SendSMS(n, fsirena);
```

```
}
```

Далі йде логічна конструкція *if(alarm)* яка відповідає за поведіння програми в випадку спрацьовування сигналізації.

```
if(key){
```

```
    lcd.print("*");
```

```
helpa++;
```

Ця конструкція відповідає за ставлення зірочки коли користувач вводить пароль.

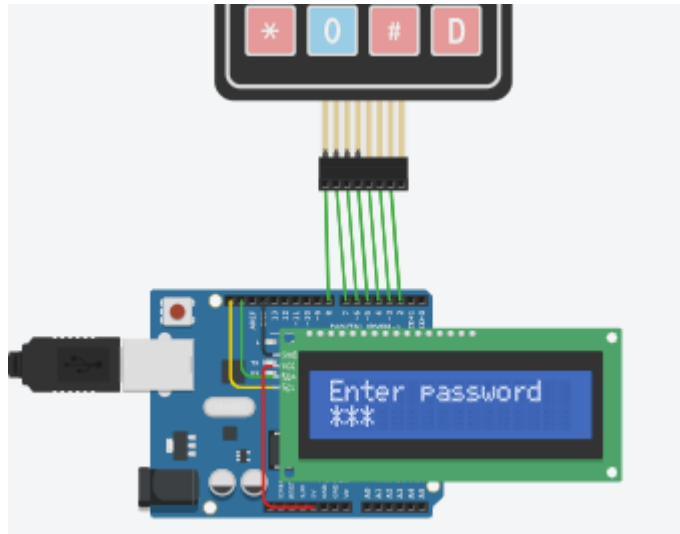


Рисунок 3.5 – Ставлення зірочки коли користувач вводить символ

Потім потрібно перевіряти чи ввів користувач правильний пароль чи ні і це доволі тяжко зробити тому що головна функція *loop* знаходиться в постійній ітерації тому прийняте рішення застосувати тип даних *bool* так як він займає найменше пам'яті, а в платі її доволі мало, для цього створено наступну логічну конструкцію:

```
if(helpa != 4){  
    if(key){  
        lcd.print("*");  
        helpa++;  
    }  
}
```

Ця конструкція відповідає за поставлення зірочок як було вказано вище.

```
if(key == pass1 && pass10 == false)  
{  
    pass10 == true;  
}  
if(key == pass2 && pass20 == false && pass10 == true)  
{  
    pass20 == true;  
}
```



```

    }
    if(key == pass3 && pass20 == true && pass30 == false)
    {
        pass30 == true;
    }
    if(key == pass4 && pass40 == false && pass30 == true)
    {
        pass40 == true;
    }
    }
    }
}

```

Кожен *pass* в конструкції відповідає за своє число в паролі і коли користувач вводить перше число, перша конструкція (*pass1*) закривається тому що *pass10* тепер *false* і так само аж до *pass4*.

```

else{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("wrong password");
    delay(5000);
    helpa = 0;
}

```

Якщо ж одне з чисел було не правильним то за допомогою команди *lcd.clear()*; все на дисплеї стирається і на екран виводиться повідомлення *wrong password* потім після затримки в 5 секунд на екрані знову з'являється повідомлення про введення паролю.

Конструкція вище відповідає тільки за роботу пристрою при спрацюванні сигналізації, коли датчик газу ловить високу концентрацію газу то *bool* змінна *alarm* стає *true*, або вірно, але якщо тривога не була піднята то змінна *alarm* буде *false* і конструкція не спрацює.

```

n1.toCharArray(n,20);

```

```
sms.SendSMS(n, sirena);
```

Посилаємо СМС про те що включена сирена

```
if(input_string=="1"){Serial.print("Postanovkanaohranu!");n1.toCharArray(
n,0); sms.SendSMS(n, smsW); sms.DeleteSMS(1); memset(n,0,20);
```

Команда "1" - Поставити на охорону

```
sensor1=1; input_string=""; char smsbuffer[160]=""; flag1=0;}
```

```
sms.DeleteSMS(1);memset(n,0,20);
```

Видаляємо СМС з сім карти.

```
else{
```

```
if(val == HIGH){
```

Змінна *val* відповідає за приймання даних з датчику газу *HIGH* означає що датчик подав інформацію на плату, а в такому випадку потрібно підняти тривогу, але іноді це могла бути тільки випадковість наприклад пролітавши біля датчика комаха, тому було зроблено логічну конструкцію, яка піднімає тривогу тільки після засікання підвищеної концентрації газу.

```
if(cou == 1){
```

```
tone (piiiPin, 1000);
```

Ця функція відповідає за ввімкнення пищалки при піднятті тривоги і вимнеться тільки після введення правильного паролю або смс повідомлення з телефону.

```
lcd.setCursor(0, 0);
```

```
lcd.print(" ");
```

```
lcd.setCursor(0, 0);
```

```
lcd.print(" ALARM!!!");
```

Тут виводиться слово *ALARM!!!* на дисплей і після затримки в 3 секунди пропадає для логічної конструкції з паролем.

```
lcd.setCursor(0, 1);
```

```
lcd.print(": ");
```

```
delay(3000);
```

```
alarm = true;
```

```
}
```

Саме тут змінна *alarm* отримує положення *true* і через це вмикається конструкція з підтвердженням паролю.

Якщо датчик газу засік зависоку концентрацію газу то використовується наступна конструкція:

```
else{  
Serial.println("Gas!");  
tone (piiiPin, 1000);
```

Пищалка вмикається але всього на 2 секунди.

```
lcd.setCursor(0, 0);  
lcd.print("          ");  
lcd.setCursor(0, 0);  
lcd.print("Gas detected!");
```

На екран виводиться *Gas detected* на 3 секунди.

```
lcd.setCursor(0, 1);  
lcd.print("          ");  
delay(3000);  
cou++;
```

Змінна *cou* збільшується на 1 щоб при наступній ітерації ця конструкція не виконувалася а виконалась конструкція *alarm*.

```
}
```

```
}
```

Далі йде код який відповідає за роботу програми в штатному режимі:

```
else{  
digitalWrite(piiiPin, LOW);
```

Строка відповідає за подання на пін пищалки 0 вольт щоб запобігти випадковому вмиканню або посторонніх шумів.

```
Serial.println("No Gas");
```

Для перевірки роботи модулю створено команду яка відправляє в порт повідомлення *No Gas*.

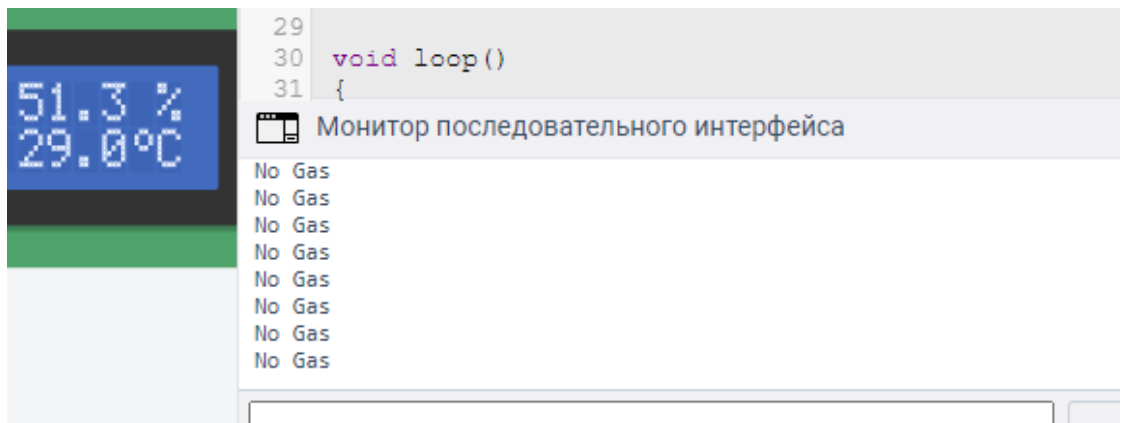


Рисунок 3.6 – Вигляд ком-порту з повідомленням *No Gas*

noTone (piiiPin);

Функція *noTone* зупиняє сигнал пищалки, нічого не робить якщо тривога не підіймалася але після тривоги необхідна для зупинки пищалки.

```
lcd.setCursor(0, 0);
```

```
//lcd.print("                ");
```

```
lcd.print("Humidity =   %");
```

```
lcd.setCursor(11, 0);
```

```
lcd.print(h, 1);
```

Виводить на першу строку екрану показник вологості.

```
lcd.setCursor(0, 1);
```

```
//lcd.print("                ");
```

```
lcd.print("Temperat =   \1C");
```

```
lcd.setCursor(11, 1);
```

```
lcd.print(t,1);
```

Виводить на другу строку дисплею показник температури.

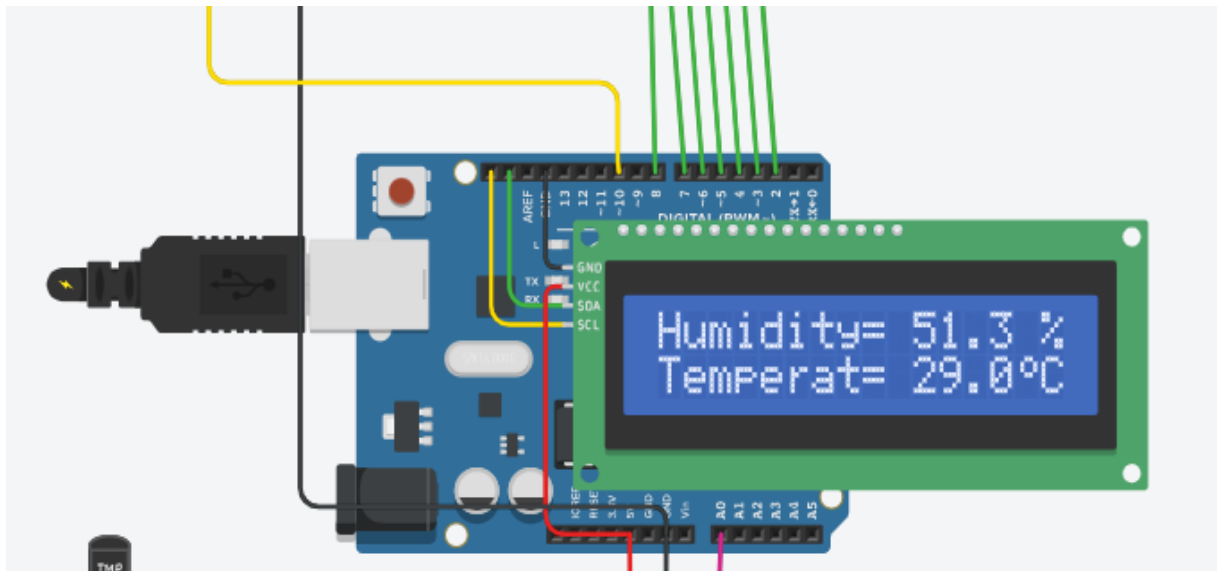


Рисунок 3.7 – Температура і вологість на дисплеї

```
}  
delay(1000);
```

Затримка програми на секунду щоб показники на дисплеї не мигали.

```
}  
}
```

3.2 Висновки до розділу

1. Розглянуто компоненти для розробки апаратного модуля – платформа *Arduino* та проміжні елементи системи.
2. Побудована схема підключення всіх складових модуля.
3. Описано етапи розробки коду в середовищі для програмування *Arduino IDE*, відображені частини коду з повним описанням всіх функцій.
4. Наведено кроки створення програмного модуля: від підготовки та налаштування до написання і компіляції коду програми.
5. Описано використані для створення проєкту утиліти командного рядка, заголовні файли та розібрані рядки коду.

РОЗДІЛ 4

КЕРІВНИЦТВО КОРИСТУВАЧА

4.1 Технічні параметри апаратного модуля “*Security_system*” та його тестування

Технічний параметр – характеристика окремої властивості об’єкта або процесу, яка допускає кількісну оцінку.

Параметри об’єкта (пристрою, системи) або процесу можуть бути зосередженими або розподіленими у площині чи просторі. Параметри у техніці за видом відношення до об’єкта (процесу) поділяються на вхідні, внутрішні і вихідні.

Вхідні (зовнішні) параметри відображають зовнішні вимоги до об’єкта. Їх величини або характер зміни з тією або іншою точністю відомі. Частина цих параметрів, що істотно впливають на стан і характеристики об’єкта, називають керувальними.

Частина вхідних параметрів, які характеризують виконувану пристроєм (процесом) функцію, відносять до функціональних параметрів. Ці параметри уже відомі в процесі проектування.

Внутрішні параметри характеризують стан і властивості самого пристрою (процесу). Їх значення визначаються або уточнюються в процесі проектування.

Вони необхідні для обґрунтування прийнятих рішень, характеристики властивостей пристрою та інших цілей.

Після завершення робіт з розробки прототипа, проведено тестування пристрою на практичному застосуванні.

Всі складові пристрою працювали справно, без жодних відхилень.

Після ввімкнення пристрою на дисплеї було показано значення з датчику

Кафедра КСУ				НАУ 22 28 52 000 ПЗ			
Виконав	Черненко Д.О.			КЕРІВНИЦТВО КОРИСТУВАЧА	Літера	Аркуш	Аркушів
Керівник	Нечипорук В.В.				Д	46	52
Консультант					СП-435Б 123		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

вологи та температури, допустиме відхилення 1 градус для датчика температури і 2 градуси для датчика вологи.

Інтервал виявлення датчиком газу концентрації в повітрі 200 - 10000 *ppm*. Після пересікання датчика значення в 7000 *ppm* на дисплеї вивелося повідомлення *Gas detected*, пищалка достатньо голосно запищить, а на телефон прийде СМС повідомлення *Sirena vkluchena!*. Після 3 секунд на дисплей виводиться напис *Enter password* зі пустим рядком знизу і мигаючим курсором.

Якщо ввести неправильний пароль на дисплей виведеться напис *wrong password*.

Якщо ввести правильний то пищання припиняться і на дисплеї з'явиться початковий екран з температурою і вологістю.

4.2 Керівництво користувача

Після підключення плати до струму, а саме або через *USB* кабель(рис.4.1 синій овал) до звичайного блоку як в мобільного телефону(або схожої альтернативи), або через провід в розетку в якій 220 вольт, але тоді потрібно підключати провід до іншого входу ардуіно (рис.4.1 червоний овал).



Рисунок 4.1 – Входи для подачі струму на платі *Arduino uno*

Для роботи *GSM* модулю потрібно засунути в нього картку на якій стоїть тариф на багато СМС повідомлень або просто лежить якась сума.

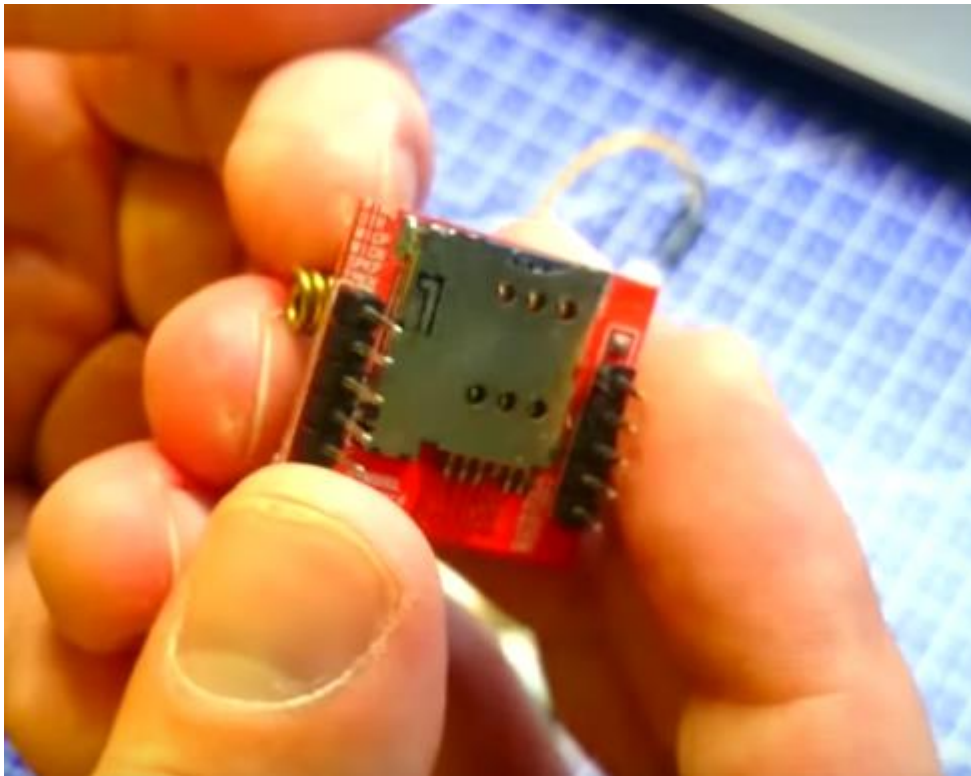


Рисунок 4.2 – Слот для сім-карти на модулі

Після того як сім-карта встановлена, з телефону на який користувач хоче отримувати повідомлення від сигналізації, потрібно зателефонувати на номер телефону сім-карти яка стоїть в модулі.

Після цього модуль запам'ятає номер з якого телефонували і буде посилати повідомлення на цей номер.

Коли плата підключена до струму автоматично вмикається дисплей і через декілька секунд виводить на екран показник температури (по Цельсію) і показник вологості (в відсотках).

Модуль буде працювати весь час поки підключений до струму.

Після того як датчик газу помітить висок концентрацію газу на дисплеї з'явиться повідомлення *Gas detected*, або датчики температури і вологи помітять завищу температуру або вологу, на дисплеї з'явиться повідомлення *High temperature* і *High humidity* відповідно, ввімкнеться зуммер а *GSM* модуль

пошле СМС повідомлення на, заданий раніше, номер телефону.

Щоб вимкнути пицалку і поставити модуль в штатний режим потрібно ввести пароль(1,2,3,4) через клавіатуру(рис.4.4).



Рисунок 4.3 – Клавіатура на якій потрібно вводити пароль

4.3 Висновки до розділу

- 1.Проведено тестування апаратного модуля.
2. Створено керівництво користувача по керуванню модулем і коротким описом принципу його роботи.

ВИСНОВОК

1. Під час виконання дипломного проекту проаналізовано сучасні програмні рішення в області проектування дистанційних систем. Отримано основні відомості про раціональне, спрощене та продуктивне створення програмно- апаратних модулів з використання платформи *Arduino*.
2. Проаналізовано технології створення апаратних модулів безпеки об'єктів інфраструктури. Обрано для реалізації даного проекту радіотехнологія, оскільки підтримує досить високу швидкість передачі і досить дешева.
3. Описано функціональні вимоги апаратного модуля, обрані технології та методи створення проекту.
4. Розроблено структуру апаратного модуля, в якій містяться деталі та компоненти програмного модулю. Пристрій має апаратну та програмну складові.
5. Проведено підбір та аналіз характеристик всіх необхідних деталей для апаратної частини.
6. Наведено діаграми варіантів використання програмного модуля.
7. Розглянуто компоненти для розробки апаратного модуля – платформа *Arduino* та проміжні елементи системи.
8. Побудовані схеми підключення всіх складових модуля та принципова схема пристрою.
9. Описано етапи розробки коду в середовищі для програмування *Arduino IDE*, відображені частини коду з повним описанням всіх функцій.
10. Наведено кроки створення програмного модуля: від підготовки та налаштування до написання і компіляції коду програми.
11. Описано використані для створення проекту утиліти командного рядка, заголовні файли та розібрані рядки коду.
12. Проведено тестування апаратного модуля та описані його технічні параметри; розглянуто всі кроки тестування апаратного модуля.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AutoDesk Tinkercad [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tinkercad.com/dashboard>
2. Стивен Прата. Язык программирования С. Лекции и упражнения, 6-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2015. — 928 с. : ил. - Парал, тит. англ.
3. Генри С. Уоррен мл. Алгоритмические трюки для программистов / Вильямс, 2014 – 512с.
4. Arduino [Електронний ресурс] / Wikipedia.com – Електрон. дані – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Arduino>
5. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П . Язык программирования C#. Классика Computers Science. 4-е изд. / Петербург: БХВ-Петербург , 2011 – 784 с.
6. MQ-2 Sensor [Електронний ресурс] – Режим доступу до ресурсу: <https://components101.com/sensors/mq2-gas-sensor>
7. Arduino IDE [Електронний ресурс] – Режим доступу до ресурсу: [Software | Arduino](#)
8. LCD Display – Fundamentals [Електронний ресурс] – Режим доступу до ресурсу: <https://electrosome.com/lcd-display-fundamentals/>
9. ГОСТ Р МЭК 60447-2000. Интерфейс человеко-машинный. Принципы приведения в действие. М., 2000. 20 с.
10. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення / Видавничий дім «Держстандарт України», 1995. – 88с
11. GSM модуль [Електронний ресурс] – Режим доступу до ресурсу: <https://www.circuitstoday.com/interface-gsm-module-with-arduino>
12. Нестеров А. Л. Проектирование АСУТП. Методическое пособие. Книга 2. –СПб. : ДЕАН, 2009. – 944
13. Техногенні небезпеки [Електронний ресурс] – Режим доступу до ресурсу:

https://spravochnick.ru/bezopasnost_zhiznedeyatelnosti/sovokupnost_i_klassifikaciya_opasnostey/tehnogennye_opasnosti/

14. Ajax systems [Электронный ресурс] – Режим доступа до ресурсу: <https://ceeua.com/uk/vendor/ajax/>