ЗМІСТ

Перелік умовних позначень, скорочень і термінів 4
Вступ 5
1 Аналітична частина 6
1.2 Опис предметної області 6
1.3 Характеристика об’єкта розробки 14
1.4 Аналіз існуючих аналогів 14
1.5 Вибір середовища розробки 17
1.4 Постановка завдання 19
2 Проектування системи 20
2.1 Вибір елементів до системи 20
2.2 Проектування системи UML-засобами 32
2.3 Схема зборки 35
3 Програмування пристрою 41
3.1 Опис функціонування системи 41
3.2 Опис алгоритму роботи 41
Висновки 51
Перелік джерел посилання 52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СМС – послуга обміну короткими текстовими повідомленнями в телекомунікаційних мережах

GSM – глобальна система мобільного зв'язку

UML – Unified Modeling Language

USB – Universal Serial Bus

Wi-Fi – технологія бездротового доступу в інтернет

Терміни

Джампери – згруповані дроти, які використовують для підключення модулів

Піни – роз’єми на платі Arduino

Скетч – програма, створена в середовищі розробки Arduino

Шілд – Shield уніфіковані сторонні модулі

Arduino − торгова марка апаратно-програмних засобів для побудови простих систем автоматики і робототехніки

Arduino IDE – середовище розробки

COM − послідовний порт

C/C ++ − мови програмування

IoT – інтернет речей

Fritzing – програмне забезпечення з відкритим кодом для віртуального моделювання електричних ланцюгів, схем і електронного обладнання

GND – пін мікроконтролера (земля)

GPIO1-GPIO16 – виходи на платі, які відповідають цифровим 0-15 Arduino

Use case – варіанти використання у мові UML

ВСТУП

Питання щодо безпеки власного житла хвилювало людину з давніх часів. І якщо раніше надійним захистом вважався потужний дверний замок, броньовані двері або злий пес, то в століття розвинених інформаційних технологій популярними стали розумні автоматизовані охоронні системи. За допомогою таких систем можна здійснювати цілодобовий контроль за будь-яким об'єктом – квартирою, котеджом, офісом або складом.

Зараз існує велика кількість таких систем, розроблених як великими компаніями, так і звичайними людьми. Такі системи бувають дуже різноманітними але виконують вони одну й ту саму роль: стоять на захисті від грабіжників.

В даному дипломному проекті буде розглянуто «розумну» систему безпеки дому, на мікроконтролері Arduino. Такий проект можна розробити самостійно, маючи чіткий план та навички в програмуванні.

Класична система безпеки будинку складається з автоматизованих блоків, які виконують такі функції:

- збирають потрібну інформацію через датчики;
- аналізують дані і приймають рішення за допомогою програмованого мікропроцесора;
- реалізовують прийняті рішення, віддаючи команди різних пристроїв.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 51
- кількість рисунків – 28
- кількість таблиць – 0
- кількість посилань – 16

1 АНАЛІТИЧНА ЧАСТИНА

1.2 Опис предметної області

1.1.1 Види охоронних систем для будинку

Домашні охоронні системи, які використовуються для забезпечення безпеки будинку можна розділити на ряд категорій:

1) системи охоронної і тривожної сигналізації. Вони являють собою комплекс пристроїв, що забезпечують контроль об'єкта по певних зонах, в яких розташовані спеціальні охоронні і функціональні датчики. При спрацьовуванні датчика сигнал від нього передається на пульт охоронної компанії, яка обслуговує цей об'єкт або на мобільний пристрій власника будинку;
2) системи контролю рівня доступу. Подібного роду системи безпеки для заміського будинку включають в свій склад електронні пристрої, що забезпечують управління приводами воріт, електронними замками та іншими виконавчими механізмами, які забезпечують доступ до певних зон контрольованого об'єкта;
3) система аудіо та відео спостереження. Такі системи безпеки для заміського будинку включають в свій склад набір аудіо пристроїв і відеокамер, за допомогою яких можна підтримувати безпеку будинку і на вулиці. Сигнал від пристроїв відео спостереження передається на пульт управління охоронної компанії або охоронця, який стежить за порядком на ввіреному йому об'єкті;
4) автономні охоронні системи. Такого типу домашня сигналізація - охоронна система, що включає до свого складу набір обладнання, яке в автоматичному режимі стежить за певними зонами в будинку за допомогою функціональних датчиків. У разі виникнення нештатних ситуацій здійснюється повідомлення господарів будинку або

охоронну службу за допомогою посилки сигналу тривоги через телефонну лінію, GSM-зв'язок або інтернет-канал;

5) система розумний будинок. За допомогою обладнання цього типу забезпечується не тільки безпеку об'єкту, що охороняється, а й здійснюється контроль стану його технологічних систем таких, як електропостачання, подача води і газу. При використанні датчиків пожежного типу така домашня сигналізація зможе забезпечувати і пожежну безпеку на об'єкті [1][1)].

1.1.2 Рівні безпеки

Щоб гарантувати максимальний рівень безпеки на об'єкті, що охороняється часто застосовуються багаторівневі системи безпеки для будинку. Вони включають в свій склад кілька рівнів, на кожному з яких використовується певне обладнання, яке фіксує ті чи інші події, пов'язані з об'єктом, що охороняється. Перший рівень захисту передбачає контроль периметра охоронюваного будинку.

Для цієї мети використовується відео спостереження за допомогою камер відкритого типу. Можуть використовуватися камери безперервної дії, які ведуть запис цілодобово, або пристрої, які володіють інфрачервоними датчиками. Такі відеокамери будуть вести запис тільки в разі потрапляння в зону дії променю якогось об'єкта.

Другий рівень системи безпеки для дачі або будинку включає до свого складу пристрою, які стежать за ситуацією на входах в будинок. До них відносяться датчики відкривання дверей і вікон, розбиття скла, вібраційні датчики, пристрої контролю руху. До перерахованих датчиків можуть встановлюватися кілька камер, що стежать за підходами до будівлі.

[1)] [1] Системы безопасности загородного дома, виды охранных систем. URL: https://bezopasnostin.ru/ohrannaya-signalizatsiya/bezopasnost-doma.html. (дата звернення 12.03.2019).

Третій рівень безпеки передбачає контроль стану всередині будинку. Такого типу системи включають в свій склад приховані відеокамери і набір інфрачервоних датчиків, що реагують на температуру об'єктів, що рухаються. Розташовуються датчики в місцях, де зберігаються найцінніші речі. Подібні системи внутрішнього контролю використовуються також і для того щоб забезпечити безпеку багатоквартирного будинку. На додаток до перерахованих пристроїв, в будинку, в різних його місцях, можуть встановлюватися і спеціальні тривожні кнопки, за допомогою яких можна сповістити про проникнення в будинок або викликати пожежну команду у разі пожару.

1.1.3 Система охоронної і тривожної сигналізації

Ця система безпеки будинку включає до свого складу набір охоронних датчиків, електронний блок управління, а також сигнальні і виконавчі пристрої. Датчики розташовуються на об'єкті в найбільш небезпечних зонах. Їх підключення до електронного блоку забезпечується за допомогою кабелів або через радіозв'язок (рис. 1).

Рисунок 1 – Схема системи охоронної і тривожної сигналізації

У разі спрацювання датчика сигнал від нього надходить на електронний блок, який обробляє його відповідно до встановленої програми. В результаті обробки сигналу формується команда, яка передається на виконавчий пристрій, який підключається до електронного блоку також за допомогою дротових магістралей або радіоканалу.

Крім виконавчих пристроїв системи безпеки цього типу можуть транслювати сигнал тривоги на пульт охоронної компанії або на мобільні пристрої власника будинку. Для передачі сигналу за допомогою мобільного зв'язку електронний блок повинен мати в своєму складі вбудований GSM-адаптер, настроєний на роботу з одним з операторів стільникового зв'язку [2][1)].

1.1.4 Система контролю рівня доступу

Ця система гарантує високу безпеку приватного будинку, використовуючи для реалізації цієї мети комплекс технічних засобів, що обмежує доступ в будинку до різних його зон. Пристрої цієї системи можуть розташовуватися на вході в будинок, при в'їзді в гараж, в деяких приміщеннях і кімнатах. Потрапити в будинок з такою електронною охороною або в одне з його приміщень, яке знаходиться під контролем, можна тільки після зняття з охорони на центральному пульті. Реалізується ця процедура через:

- введення коду на спеціальному електронному блоці;
- застосування відповідного програмованого чіпу;
- використання магнітних карток доступу;
- систему розпізнавання відбитку пальця, сітківки ока і ін.

Система контролю доступу може функціонувати навіть коли господарі знаходяться дома [3][1)].

[1)] [2] Система ограниченного доступа вашего дома. URL: https://sovtehnika.ru/sistema-ogranichennogo-dostupa-vashego-doma.html. (дата звернення 12.03.2019).

[1)] [3] Система контролю доступу. URL: https://leater.com/ua/services/sistema-kontrolyu-dostupu.html. (дата звернення 13.03.2019).

В цьому випадку може обмежуватися доступ до деяких кімнат, наприклад до тих, де зберігаються цінні папери або речі, до майстерні, де може поранитися дитина та ін. (рис. 2).

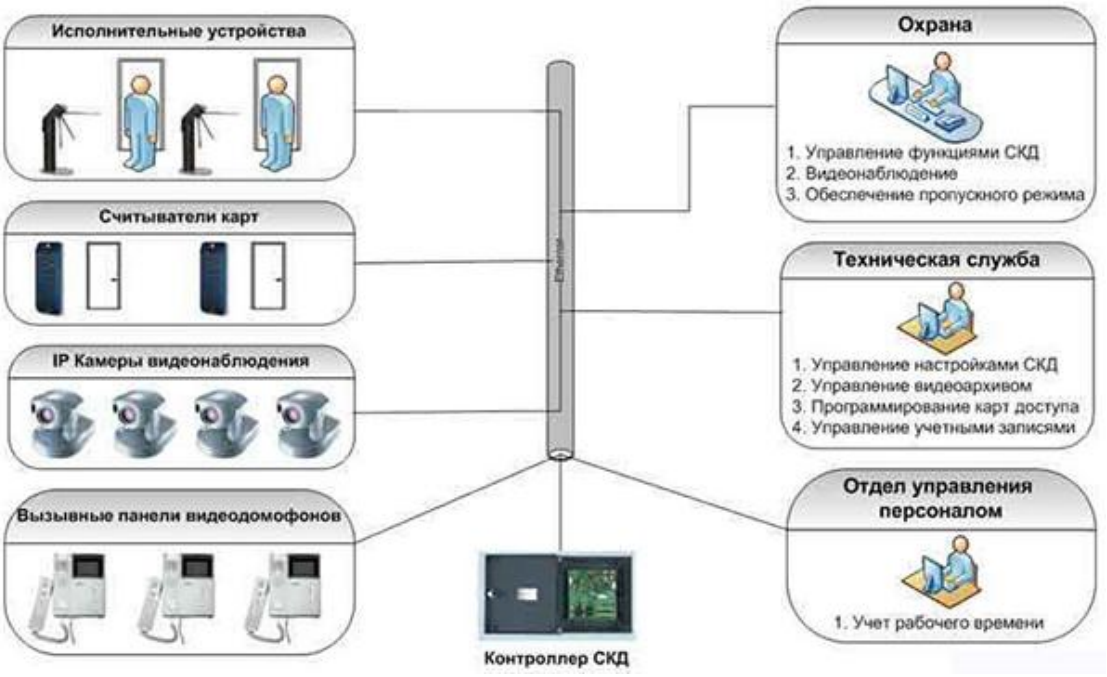

Рисунок 2 – Схема системи контролю доступу

1.1.5 Система аудіо та відео спостереження

Пристрої аудіо та відео спостереження користуються високою популярністю при створенні багатофункціональної системи безпеки. Наявність цього обладнання дозволяє візуально стежити за ситуацією, яка відбувається в будинку або навколо нього.

За допомогою таких систем можна своєчасно реагувати на проникнення злодія, а також мати незаперечні докази крадіжки конкретною людиною, які зафіксовані за допомогою відео апаратури.

У житловий будинок можуть встановлюватися різні типи камер, включаючи як приховані, так і відкриті пристрої [4][1)].

Вибираючи камеру для системи безпеки важливо враховувати, де вона буде розташовуватися - всередині приміщення або на вулиці. Другий варіант камер повинен мати відповідний захист від впливу зовнішнього середовища (рис. 3).

Для того щоб ефективно користуватися системами відео спостереження слід вибирати камери з вбудованою інфрачервоною підсвіткою. В такому випадку буде можливість фіксації порушника на досить великій відстані в темний час доби.

Підвищити ефективність роботи системи відео спостереження дозволять також і камери, які володіють поворотним механізмом. В такому випадку істотно збільшується периметр їх огляду [5][2)].

Рисунок 3 – Схема систем відео спостереження

[1)] [4] Відеонагляд. URL: http://tdim.com.ua/ua/systemy-bezpeky/videonahliad.html. (дата звернення 14.03.2019).

[2)] [5] Скрытое видео и аудіо наблюдение. URL: https://guard-lviv.com.ua/info/skrytoe-video-i-audio-nablyudenie. (дата звернення 14.03.2019).

1.1.6 Автономні охоронні системи

Функціонування цих систем безпеки для будинку аналогічно системам охоронної сигналізації. Їх особливістю є те, що всі функції виконуються повністю в автоматичному режимі, а передача сигналів можлива на великі відстані через GSM-зв'язок (рис. 4).

Рисунок 4 – Приклад автономної сигналізації

Завдяки використанню високоякісних автономних джерел живлення такі системи можуть тривалий час функціонувати навіть без доступу до побутової електричної мережі. Тому, навіть повністю знеструмлений об'єкт, буде завжди знаходитися під пильною увагою електронного охоронця, який в

будь-яку хвилину зможе попередити про проникнення сторонніх в будинок [6][1)].

1.1.7 Системи розумний будинок

Такі електронні комплекси покликані допомогти власникові будинку в його експлуатації (рис. 5). З їх допомогою підтримується не тільки необхідна безпека дачі чи будинку, а й забезпечується контроль і управління подачею води і газу в будинок, підтримується робота опалювальних систем та кондиціювання, дистанційно можна керувати різним обладнанням, яке підключене до системи розумний будинок [7][1)].

Ще однією перевагою є підтримка роботи з системами, що забезпечують пожежну безпеку будинку.

Рисунок 5 – Схема систем розумного будинку

[1)] [6] Автономные охранные сигнализации. URL: https://www.forter.com.ua/news-and-articles/avtonomnye-ohrannye-signalizatsii-kak-eto-rabotaet/. (дата звернення 14.03.2019).

[1)] [7] Розумний будинок. URL: https://www.smarthouse.ua/ua/umnyj_dom.html. (дата звернення 14.03.2019).

Наявність величезного переліку різного устаткування дозволяє створювати багато форматні конфігурації охоронних систем приватних будинків, котеджів, дач.

Залежно від особливостей об'єкта, що охороняється, а також фінансових можливостей власника, можна побудувати багаторівневий захист, який не зможе пройти жоден зловмисник, якими б засобами він не користувався.

1.3 Характеристика об’єкта розробки

Питання щодо безпеки нерухомості є дуже актуальним. Але ціна безпеки іноді досить висока і не кожен має змогу придбати навіть найпростішу систему безпеки для свого будинку. Але є і альтернативний варіант – розробити систему безпеки власноруч.

Розроблений програмний продукт – це розумна система безпеки будинку, яка реалізована на платформі Arduino.

Даний програмний продукт буде складатися з головного контролеру – «мозку» системи, який буде взаємодіяти з користувачем та різноманітних датчиків і сенсорів.

Система буде взаємодіяти з власником за допомогою СМС команд, які буде обробляти GSM модуль.

1.4 Аналіз існуючих аналогів

Існує величезна кількість різноманітних систем захисту приватної власності. Найрозвинутіші системи розробляють величезні компанії, вони вкладають в розробку величезну купу грошей. Як наслідок вартість таких систем безпеки дуже висока і не кожен може дозволити їх придбати.

З іншого боку існує багато менш дорогих рішень але вони не можуть похвалитися високою якістю.

Найпопулярнішою компанією в Європі з розробки систем безпеки будинків стала компанія «Ajax Systems», яка була заснована у 2011 році.

Це українська компанія, яка представлена більш ніж у 70 країнах і має близько 100 000 об'єктів під охороною по всьому світі. Устаткування компанії заслужило популярність не тільки завдяки розумній ціні, але і завдяки високій якості, приємному дизайну і широкому асортиментом охоронних систем [8][1)].

На даний момент Ajax має приблизно 15 різних пристроїв, що підключаються до єдиного хабу Ajax Hub. Це датчики руху, відкриття дверей і вікон, удару і нахилу, розбиття скла, протікання, диму, високої температури і чадного газу. Плюс є комбіновані датчики і реле управління живленням електроприладів.

Всі датчики бездротові, дальність прийому-передачі сигналу між датчиком і хабом досягає аж до 2 км при умовах прямої видимості. Від батареї датчики працюють аж до 7 років. Систему безпеки можна масштабувати, додавати або прибирати пристрою. Максимальне число датчиків, які підтримує ця система – 100. Цього цілком достатньо навіть для великого підприємства.

Це не звичайне рішення, а професійна система безпеки, стійка до спроб злому. Вона пройшла незалежну перевірку і отримала кілька європейських сертифікатів.

Наступна система Pitbull Alarm. Дане обладнання популярно завдяки простоті установки і відмінному асортименту. Варто сказати, що незважаючи на те, що компанія зареєстрована в Литві, вартість обладнання цілком адекватна.

Pitbull Alarm має на борту вбудований GSM / GPRS модуль і датчик руху. У разі тривоги, система сповістить господаря або охоронну компанію за стандартними міжнародними протоколами.

[1)] [8] Ajax Systems. URL: https://ajax.systems/. (дата звернення 14.03.2019).

Охоронна система Pitbull Alarm підтримує мобільні додатки і хмарний сервіс Eldes Security app [9][1)].

Jablotron – обладнання Jablotron 100. Розробки даної компанії займають стабільну позицію на ринку вже досить довгий час. Важливо знати, що дизайн датчиків практично не змінився, власне як і "начинка".

Крім іншого, для установки сигналізації потрібно обов'язково залучати фахівців, оскільки його майже неможливо встановити самостійно. Якщо говорити про ціни - це обладнання є одним з найдорожчих на ринку. Jablotron на Українському ринку з 2013 року [10][2)].

ITV – розробка 4064P, найдешевша і найпростіша в плані дизайну бездротова охоронна сигналізація. Виробництво зареєстровано в Україні, але на відміну від Ajax не може похвалитися високою якістю [11][3)].

Компанія Integrated Technical Vision Ltd. почала виробляти зчитувачі, контролери, програмне забезпечення та супутнє обладнання для систем контролю і управління доступом в 1992 році. Всі прилади розроблені і вироблені в Україні. Сертифіковані в Україні і ЄС.

Підведемо підсумки. Перш за все, необхідно знати, що все перераховане вище обладнання може працювати через хмарні сервіси і мобільні додатки. Крім того, дані системи можуть продовжити роботу в автономному режимі і підключатися до пультів охоронних служб.

З огляду на той факт, що вартість бездротової системи сигналізації порівняно вище дротових систем, а клас захисту нижче, вони все-таки є найбільш гарним варіантом для захисту приватного будинку або квартири.

---

1) [9] Pitbull Alarm Intrusion Alarm Panel, Eldes Alarms. URL: https://eldesalarms.com/product/pitbull-alarm/. (дата звернення 15.03.2019).

2) [10] Умная сигнализация Jablotron. URL: https://www.jablotron.com/ru/. (дата звернення 15.03.2019).

3) [11] Сигнализация в дом. Развитие технологий и тренды 2018. URL: https://krok-ua.com/signalizacija-dlja-doma-i-kvartiry-novye-trendy-razvitija-v-2018-godu/. (дата звернення 15.03.2019).

1.5 Вибір середовища розробки

Arduino – інтегроване середовище для розробки програмного забезпечення. Включає в себе вбудований редактор коду і спеціальні інструменти для прошивки робототехніки та автоматики.

Програми, що створюються в середовищі розробки Arduino, іноді ще називають скетчами. Скетчі пишуться в текстовому редакторі і зберігаються в файлах з розширенням .ino.

Вбудований текстовий редактор має стандартні інструменти копіювання, вставки, пошуку і заміни тексту. Область повідомлень у вікні програми є, свого роду, зворотним зв'язком для користувача і інформує його про події (в тому числі і про помилки), що виникають в процесі запису або експорту написаного коду [12][1)].

Консоль відображає у вигляді тексту потік вихідних даних середовища Arduino, включаючи всі повідомлення про помилки та ін.

У нижньому правому куті вікна програми показується модель поточної плати і послідовний порт, до якого вона підключена. Кнопки на панелі інструментів призначені для створення, відкриття, збереження і прошивки програм в пристрій. Окрема кнопка запускає програму Serial Monitor.

Для того щоб прошити програму в контролер, необхідно правильно обрати плату і послідовний порт в меню Tools > Board і Tools > SerialPort відповідно.

Після вибору використовуваного порту і плати, необхідно натиснути кнопку Upload на панелі інструментів або вибрати пункт Upload з меню File. Після цього відбудеться скидання Arduinoi почнеться процес завантаження програми в пам'ять контролера. Після завершення процесу прошивки, програма видасть відповідне повідомлення або помилку.

[1)] [12] Среда разработки Arduino. URL: http://arduino.ru/Arduino_environment. (дата звернення 15.03.2019).

Завантаження програми в сам модуль здійснюється за допомогою завантажувача – невеликої програми, яка вшита в пам'ять мікроконтролера. Вона дозволяє завантажувати в нього код без зовнішніх апаратних засобів.

Завантажувач активізується на кілька секунд після перезавантаження пристрою, після чого він запускає на виконання останній завантажений в контролер скетч.

Можливості Arduino IDE:

- сумісність з будь-платою Arduino;
- використовує мову програмування C ++;
- можливість працювати з декількома документами;
- функції збереження, експорту, пошуку і заміни текстового коду;
- багато довідкової інформації на офіційному сайті розробника;
- містить необхідні параметри для управління зовнішніми пристроями;
- взаємодія з MacromediaFlash, SuperCollider, PureData, Max / MSP і так далі;
- наявність текстового редактора, компілятора і модуля для установки нових прошивок плати.

Переваги:

- зручні елементи управління;
- середовище розробки написана на Java;
- невисокі системні вимоги;
- доступне меню російською мовою;
- швидке завантаження скетчу в Arduino;
- володіє відкритим вихідним кодом;
- можливість створювати прошивки для мікроконтролерів;
- велика кількість готових прикладів для початківців програмістів;
- сумісність середовища розробки Arduino IDE з ОС Windows, Linux, Mac OS (32 і 64 bit).

Недоліки:

- можливі збої при завантаженні прошивки на Arduino Uno R2;
- послідовний порт при прошивці необхідно вибирати вручну.

1.4 Постановка завдання

Метою даної роботи є проектування та розробка розумної системи безпеки будинку на базі Arduino. Для розробки системи і досягнення поставленої мети необхідне рішення наступних завдань:

- дослідження і аналіз компонентів, які дозволяють реалізувати поставлену мету;
- проектування системи;
- написання програмного коду.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Вибір елементів до системи

Arduino – це платформа для розробки електронних пристроїв з автоматичним, напівавтоматичним або ручним керуванням. Вона виконана за принципом конструктора з чітко визначеними правилами взаємодії між елементами. Система є відкритою, що дозволяє стороннім виробникам брати участь в її розвитку.

Платформа Arduino гарна саме тим, що не замикається на певному виробнику, а дозволяє самому підібрати відповідні компоненти. Їх вибір величезний, тому можна реалізувати практично будь-які задуми.

Крім різноманіття пристроїв, що підключаються, варіативності додає середовище програмування, реалізоване на мові C++. Користувач може не тільки скористатися створеними бібліотеками, а й сам запрограмувати реакцію компонентів системи на виникаючі події [13][1)].

Материнська плата об'єднує в собі такі елементи:

- мікроконтролер (процесор). Основне його призначення - видавати і вимірювати напругу в портах в діапазоні 0-5 або 0-3.3 В, запам'ятовувати дані і робити обчислення;
- програматор. За допомогою цього пристрою в пам'ять мікроконтролера записують програму, згідно з якою буде працювати "розумний будинок". До комп'ютера, планшета, смартфона або іншого пристрою його підключають за допомогою USB-інтерфейсу;
- стабілізатор напруги. Необхідно пристрій на 5 вольт, потрібно для живлення всієї системи.

На материнській платі мікроконтролера існують два види портів:

- цифрові, які позначені на платі буквами "d";

[1)] [13] Arduino ТОП-3 графических сред программирования CHIP. URL: https://ichip.ru/arduino-top-3-graficheskikh-sred-programmirovaniya.html. (дата звернення 17.03.2019).

- аналогові, які позначені літерою "a".

Завдяки їм мікроконтролер здійснює зв'язок з підключеними пристроями. Будь-порт може працювати як на отримання сигналу, так і на його віддачу. Цифрові порти з позначкою "pwm" призначені для введення і виведення сигналу типу ШИМ (широтно-імпульсна модуляція).

Кожен мікропроцесор оснащений трьома видами пам'яті:

- flash Memory. Основна пам'ять, де зберігається код програми управління системою. Незначну її частину (3-12%) займає вшита програма завантаження (bootloader);
- sRAM. Оперативна пам'ять, де зберігаються тимчасові дані, необхідні при роботі програми. Відрізняється високою швидкістю роботи;
- EEPROM. Більш повільна пам'ять, де також можна зберігати дані.

Основна відмінність видів пам'яті для зберігання даних полягає в тому, що при виключенні електроенергії інформація, яка записана в SRAM, втрачається, а в EEPROM залишається. Але у енергонезалежного типу є і недолік – обмежене число циклів запису. Це потрібно пам'ятати при створенні власних додатків [13][1)].

Всі модулі, які будуть підключені до плати, мають як мінімум три виходи. Два з них – дроти живлення, а також напруга 5 або 3.3 В. Третій провід є логічним. По ньому йде передача даних до порту. Для підключення модулів використовують спеціальні згруповані по 3 штуки дроти, які іноді називають «джамперами».

Для збільшення можливостей материнських плат використовують Шилди (Shields). Ці додаткові пристрої розширюють функціонал основної плати. Вони виготовляються під конкретний форм-фактор, що відрізняє їх від модулів, які підключають до портів. Шилд коштує дорожче модулів, однак робота

[1)] [13] Arduino ТОП-3 графических сред программирования CHIP. URL: https://ichip.ru/arduino-top-3-graficheskikh-sred-programmirovaniya.html. (дата звернення 17.03.2019).

з ним простіше. Також вони забезпечені готовими бібліотеками з кодом, що прискорює розробку власних програм управління для "розумного будинку".

Підключення модулів від сторонніх виробників і можливість роботи з ними, використовуючи вбудовану мову програмування – основна перевага відкритої системи Arduino в порівнянні з "фірмовими" рішеннями для "розумного будинку". Головне, щоб модулі мали опис одержуваних або переданих сигналів.

Способи отримання інформації: введення інформації може бути здійснений через цифровий та аналоговий порти. Це залежить від типу кнопки або датчика, який отримує інформацію і транслює її на плату.

Сигнал до мікропроцесора може бути посланий людиною, який використовує для цього два способи: натискання кнопки (клавіші). Логічний провід в цьому випадку йде до цифрового порту, які отримує значення "0" в разі відпущеної кнопки і "1" в разі її натискання.

Обертання ковпачка поворотного потенціометра (резистора) або зрушення важеля движкового. В цьому випадку логічний провід йде до аналогового порту. Напруга проходить через аналогово-цифровий перетворювач, після чого дані надходять до мікропроцесора.

Кнопки використовують для старту якої-небудь події, наприклад, включення і виключення світла, опалення або вентиляції. Поворотні ручки застосовують для зміни інтенсивності – збільшення або зменшення яскравості світла, гучності звуку або швидкості обертання лопатей вентилятора.

Для автоматичного визначення параметрів середовища або походження якої-небудь події використовують датчики. Для роботи "розумного будинку" найбільш затребувані наступні їх різновиди:

- датчик звуку. Цифрові варіанти цього пристрою використовують для активації якої-небудь події за допомогою бавовни або подачі голосу. Аналогові моделі дозволяють розпізнавати і обробляти звук;

- датчик світла. Ці прилади можуть працювати як у видимому, так і в інфрачервоному діапазоні. Останні можуть бути застосовані в якості системи оповіщення про пожежу;
- датчик температури. Для дому та вулиці використовують різні моделі, так як зовнішні краще захищені від впливу вологи. Є також виносні пристрої на дроті;
- датчик вологості повітря. Для приміщення піддійте модель DHT11, а для вулиці – дорожча DHT22. Обидва пристрої також можуть давати і показання температури. Підключаються до цифрового порту;
- датчик тиску повітря. Для спільної роботи з платами Arduino добре зарекомендували себе аналогові барометри фірми Bosh: bmp180, bmp280. Вони також вимірюють температуру. Модель bme280 можна назвати метеостанцією, так як вона видає додатково ще і значення вологості;
- датчики руху і присутності. Їх використовують в охоронних цілях або для автоматичного включення світла;
- датчик струму. Їх застосовують для виявлення непрацюючих електроприладів (перегорілих ламп) або для аналізу напруги, щоб не допустити перевантаження;
- датчик витоку газу. Застосовується для виявлення і реагування на підвищену концентрацію пропану;
- датчик вуглекислого газу. Його використовують для визначення концентрації вуглекислоти в житлових кімнатах і в спеціальних приміщеннях, таких як винні погреби, де відбувається бродіння.

Існує ще багато різних датчиків під специфічні завдання, наприклад для вимірювання ваги, швидкості течії води, відстані, вологості ґрунту і т.д.

Багато сенсори і датчики можна зробити самостійно, використовуючи більш прості компоненти. Це обійдеться дешевше. Але, на відміну від застосування серійних пристроїв, доведеться витратити час на калібрування.

Крім збору та аналізу інформації "розумний будинок" повинен реагувати на виникаючі події. Присутність на сучасних побутових приладах просунутої електроніки дозволяє звертатися до них безпосередньо, використовуючи Wi-Fi, GPRS або Ethernet. Зазвичай, для систем Arduino реалізують комутацію мікропроцесора і високотехнологічних пристроїв за допомогою Wi-Fi.

Крім "спілкування" з комп'ютеризованими приладами часто виникають завдання, пов'язані з виконанням будь-яких механічних дій. Наприклад, до плати можна підключити сервопривід або невеликий редуктор, який буде від неї живиться.

У разі необхідності підключення потужних пристроїв, що працюють від зовнішнього джерела живлення, використовують два варіанти:

– включення в електричний ланцюг електромагнітного або твердотільного реле. Це пристрій замикає й розмикає один з проводів по команді, що надходить від мікропроцесора. Основна їх характеристика – максимально допустима сила струму (наприклад, 40 А), яка може проходити через цей прилад;
– підключення силового ключа – мосфети для постійного струму і симистора для змінного. Вони володіють меншим значенням допустимої сили струму (5-15 А), але можуть плавно збільшувати навантаження. Саме для цього на платах передбачені ШІМ-порти. Це властивість використовують при регулюванні яскравості освітлення, швидкості обертання вентиляторів і т.д.

2.1.1 Комплектуючі

Arduino Piranha UNO – ця плата є повним аналогом Arduino UNO R3, але на відміну від Arduino UNO R3 на платі Piranha UNO R3 використовуються більш потужні стабілізатори напруги для шин живлення на 5В і 3.3В,

що призводить до більш стабільної роботи більшості модулів (рис.6). Приклад розпіновки плати представлено на рис. 7.

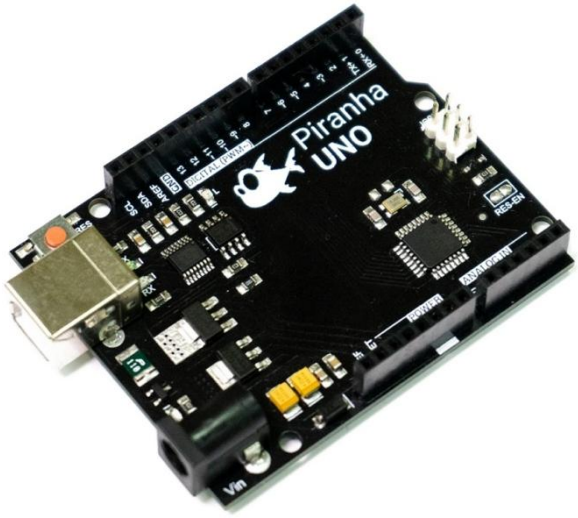

Рисунок 6 – Монтажна реалізація елементної бази плати Arduino Piranha Uno

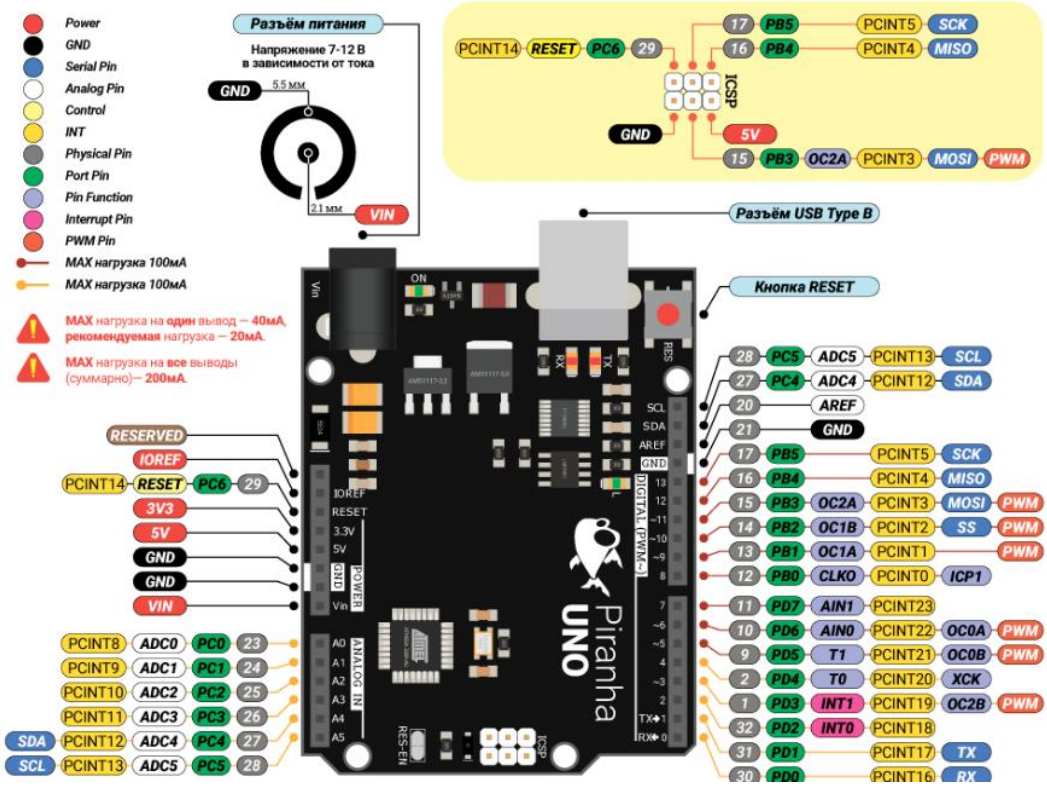

Рисунок 7 – Розпіновка плати Arduino

GSM/GPRS Shield A6 – дозволяє працювати в мережах стільникового зв'язку за технологіями GSM / GPRS для прийому / передачі даних, відправлення / отримання SMS і здійснення телефонного голосового зв'язку (рис. 8).

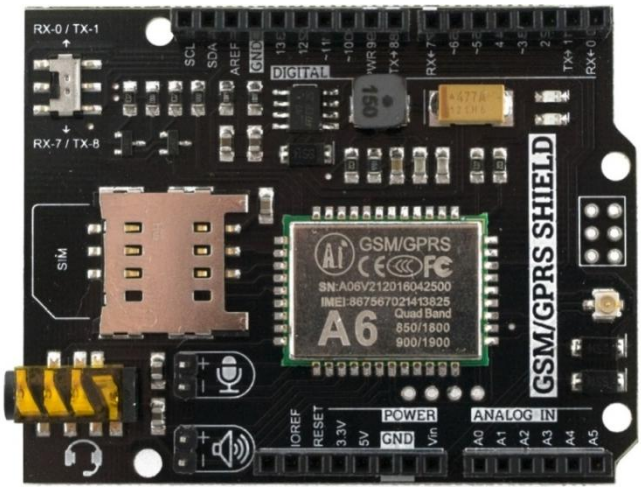

Рисунок 8 – Монтажна реалізація елементної бази плати GSM/GPRS Shield A6

Battery Shield встановлюється на Arduino, забезпечуючи її живленням як звичайний акумулятор забезпечує живленням смартфон, планшет і т.д. Якщо до шин живлення Arduino підключені інші пристрої, вони також отримають живлення від Battery Shield (рис. 9).

Рисунок 9 – Монтажна реалізація Battery Shield

Це плата розширення, яка спрощує процес підключення модулів до Arduino. Використання Trema Shield (рис. 10) позбавляє від необхідності пайки проводів при підключенні декількох модулів до Arduino, спрощує процес створення пристроїв.

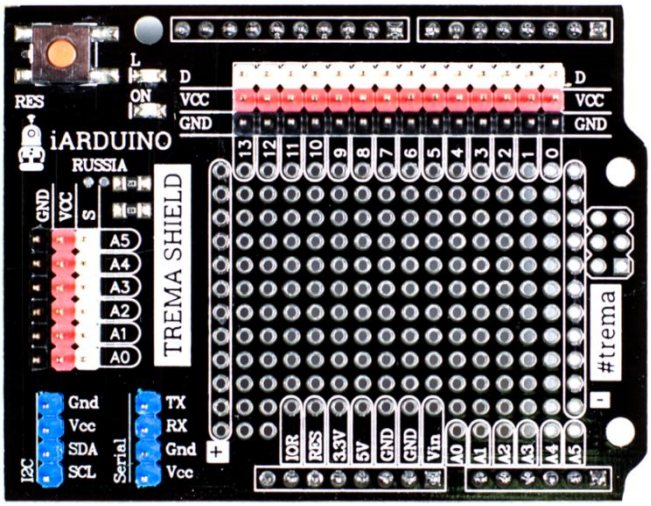

Рисунок 10 – Монтажна реалізація Trema Shield

Trema-модуль цифровий термометр: вимірює температуру навколишнього середовища. Цифровий сигнал стійкіший від наведень, на відміну від аналогового датчика температури (рис. 11).

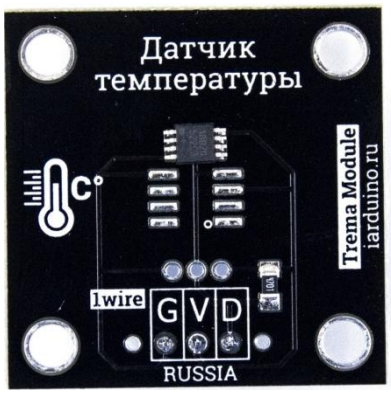

Рисунок 11 – Монтажна реалізація елементної бази цифрового термометру

Цифровий термометр (герметичний): датчик температури DS18B20 – дозволить вимірювати температуру навколишнього середовища (рис. 12).

Рисунок 12 – Цифровий термометр DS18B20

Trema-модуль – датчик газу MQ-2 (широкого спектру): даний модуль здатний визначати концентрацію широкого спектра газів в повітрі (природні гази, вуглекислий і чадний газ, вуглеводні, дим) (рис. 13).

Рисунок 13 – Монтажна реалізація елементної бази датчик газу MQ-2

ИК-датчик руху HC-SR501: він дозволяє виявити рух людей в контрольованій зоні за рахунок випромінюваного людиною тепла (рис. 14).

Рисунок 14 – ИК-датчик руху HC-SR501

Магнітний сповіщувач (геркон): це перемикач і магніт, поміщені в пластиковий корпус. Робота за принципом "замкнутий контакт – розімкнений контакт", сповіщувач дозволяє використовувати його в широкому колі завдань: контроль відкриття дверей, лічильники спрацьовувань / швидкості / частоти і т.д. (рис. 15).

Рисунок 15 – Магнітний сповіщувач

2.1.2 Бібліотеки

Бібліотека в Arduino – це програмний код в зовнішніх файлах, які можна встановити і підключити до вашого скетчу. У бібліотеці зберігаються різні методи і структури даних, які потрібні для спрощення роботи з датчиками, індикаторами, модулями та іншими компонентами. Використання бібліотек істотно спрощує роботу над проектами, тому що можна зосередитися на основною логікою програми, не витрачаючи час на безліч дрібниць. Сьогодні величезна кількість бібліотек викладено в інтернеті, де їх можна легко завантажити, причому абсолютно безкоштовно.

З точки зору файлової системи бібліотека являє собою каталог, який містить певні папки. Під час компіляції і збірки проекту Arduino IDE автоматично включає в код ті класи, структури даних і методи з бібліотек, які включені і використовуються в скетчі. Таким чином, єдине, що нам потрібно зробити - це виставити в свій код відповідні інструкції, попередньо переконавшись, що потрібна бібліотека встановлена.

Щоб почати користуватися бібліотекою, потрібно на початку програми включити заголовний h файл директивою include. Наприклад, підключення бібліотеки LiquidCrystal.h буде виглядати так: #include <LiquidCrystal.h>

Кожна бібліотека повинна містити як мінімум 2 файли – це заголовний файл з розширенням .h і файл з вихідним кодом з розширенням .cpp. В заголовки знаходяться опис класу, константи і змінні. У другому файлі містяться коди методів. Крім двох основних файлів може міститися текстовий документи keywords.txt і папка examples з кодами прикладів використання бібліотеки. Файли h і cpp не обов'язково лежать в корені [12][1)].

Встановити бібліотеку можна двома способами – за допомогою засобів Arduino IDE і вручну.

[1)] [12] Среда разработки Arduino. URL: http://arduino.ru/Arduino_environment. (дата звернення 15.03.2019).

Щоб встановити бібліотеку за допомогою засобів Arduino IDE потрібно зайти в меню Скетч – Підключити бібліотеку – Додати .Zip бібліотеку (рис. 16).

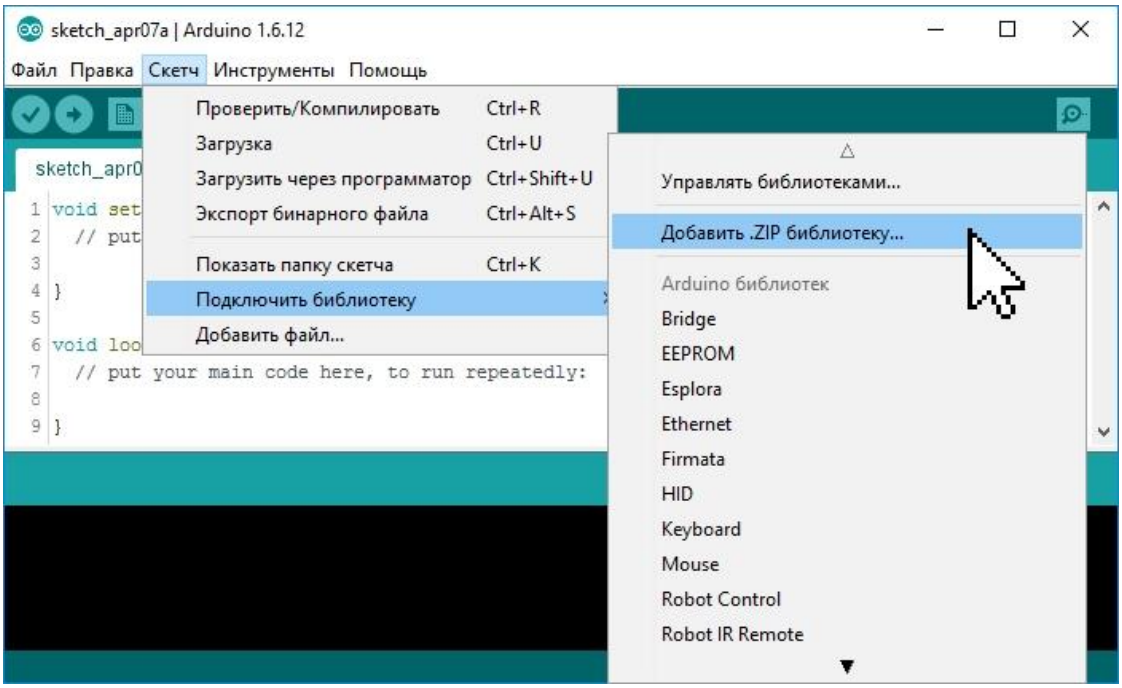

Рисунок 16 – Додавання бібліотеки в середовищі розробки Arduino

Для того щоб підключити бібліотеку, потрібно написати всього один рядок на початку скетчу: "#include<файл.h>"

Потрібні бібліотеки:

- iarduino_GSM – бібліотека для роботи з GSM/GPRS Shield;
- OneWhire – бібліотека для роботи з датчиками фірмиDallas;
- DallasTemperature – для роботи с цифровими термометрами DS18B20;
- SoftwareSerial.

2.2 Проектування системи UML-засобами

UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення.

UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю [14][1)].

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Ця діаграма є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання [15][2)].

[1)] [14] Основы UML – диаграммы использования (use-case). URL: https://proprof.com/archives/2594. (дата звернення 23.04.2019).

[2)] [15] UML UseCaseDiagramTutorial. URL: https://www.lucidchart.com/pages/uml-use-case-diagram. (дата звернення 23.04.2019).

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання.

Варіант використання (англ. Use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Так, для об’єкту розробки діаграма прецедентів буде мати наступний вигляд (рис. 17):

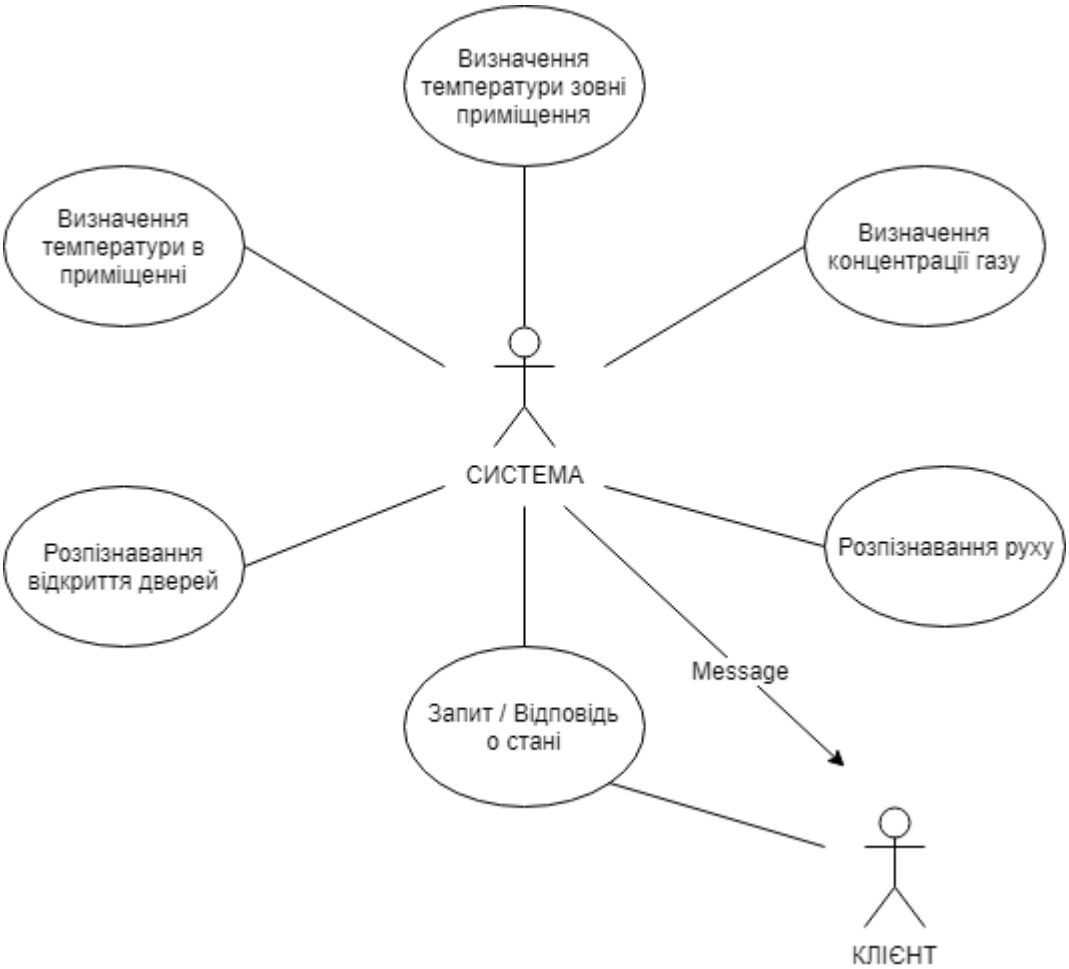

Рисунок 17 – Діаграма USE-CASE

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії.

Подібно до цього, виконання діяльності є виконанням окремої діяльності, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності.

Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку [16][1)].

Ці діаграми широко використовуються в описі поведінки, що включає велику кількість паралельних процесів. Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції.

Таким чином, діаграму діяльності можна вважати приватним випадком діаграми станів

Основним напрямком використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно надати алгоритми їх виконання.

Наступна діаграма демонструє основні етапи розробки системи безпеки будинку (рис. 18):

[1)] [16] Моделирование на UML. URL: http://book.uml3.ru/. (дата звернення 26.04.2019).

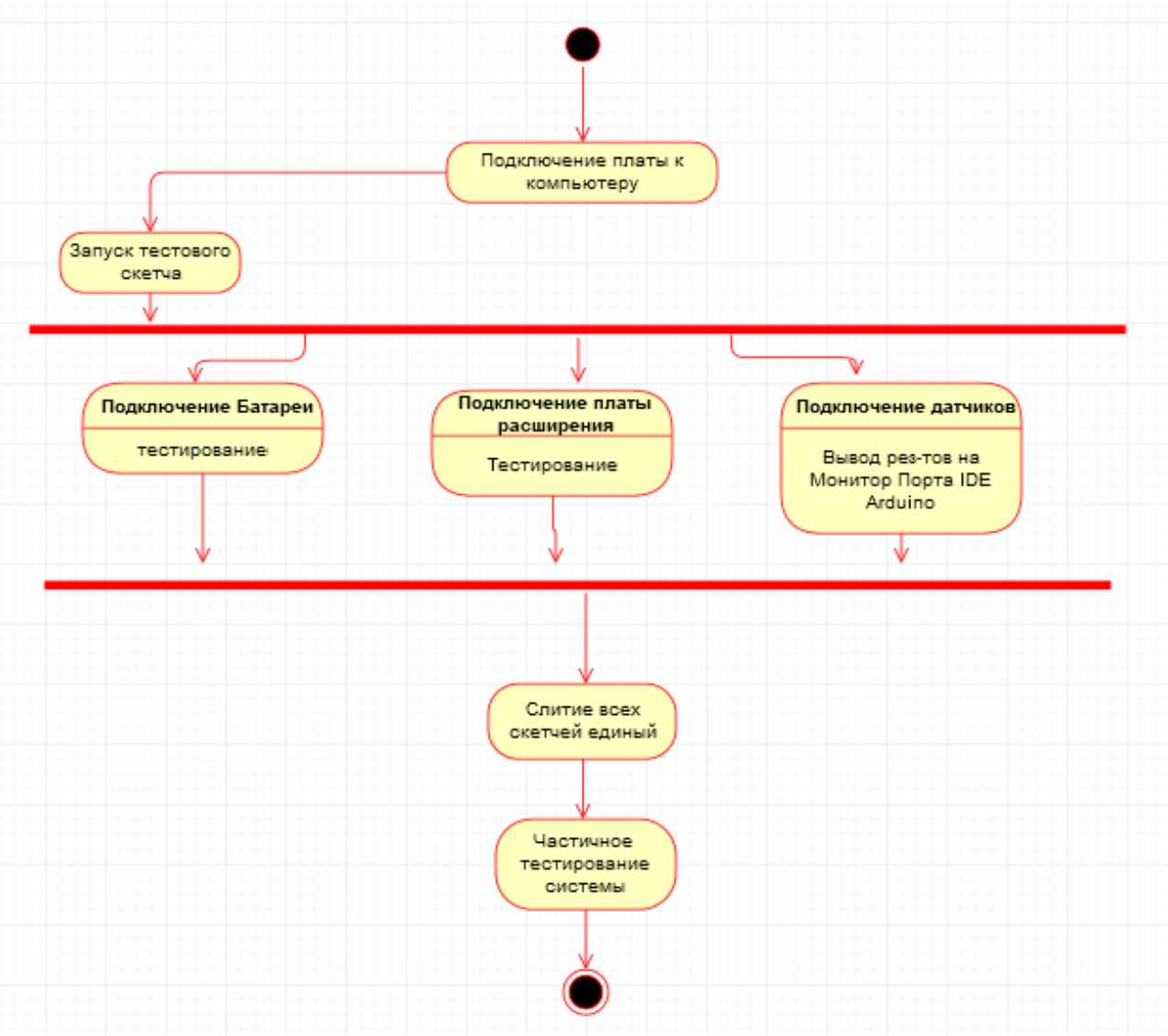

Рисунок 18 – Діаграма діяльності процесу розробки алгоритму роботи системи

2.3 Схема зборки

Основним мікроконтролером виступає Arduino Piranha Uno (рис. 19).

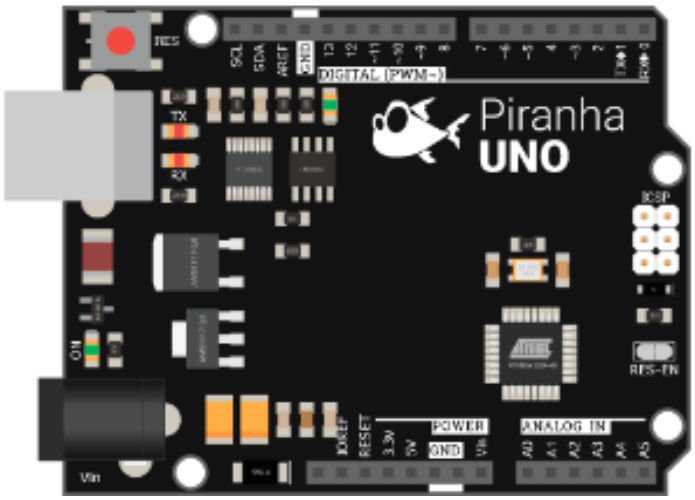

Рисунок 19 – Монтажна реалізація елементної бази плати Arduino Piranha Uno

Встановлюємо Battery Shield на Arduino Piranha UNO (рис. 20):

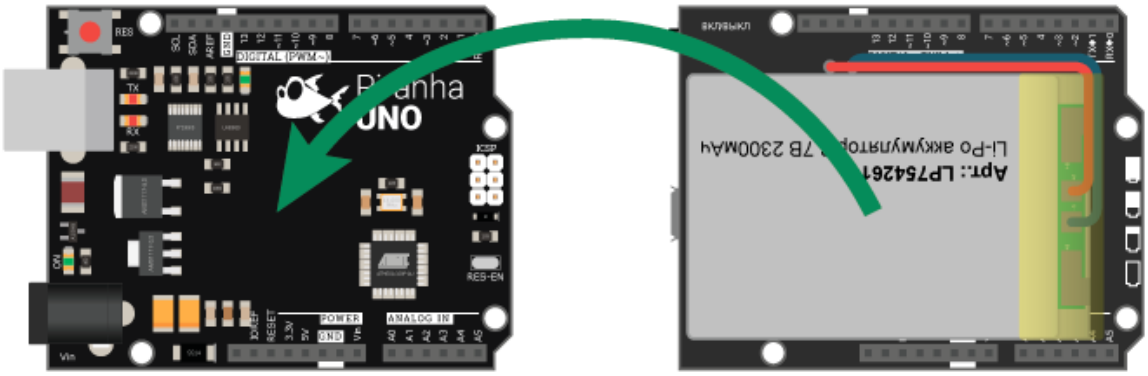

Рисунок 20 – Додавання елементу Battery Shield до Arduino Piranha UNO

Принцип підключення елементу живлення до макетної плати Arduino Piranha UNO аналогічен як і для елементу Battery Shield. Схема підключення розроблена за допомогою програми Fritzing і представлено на рис. 21:

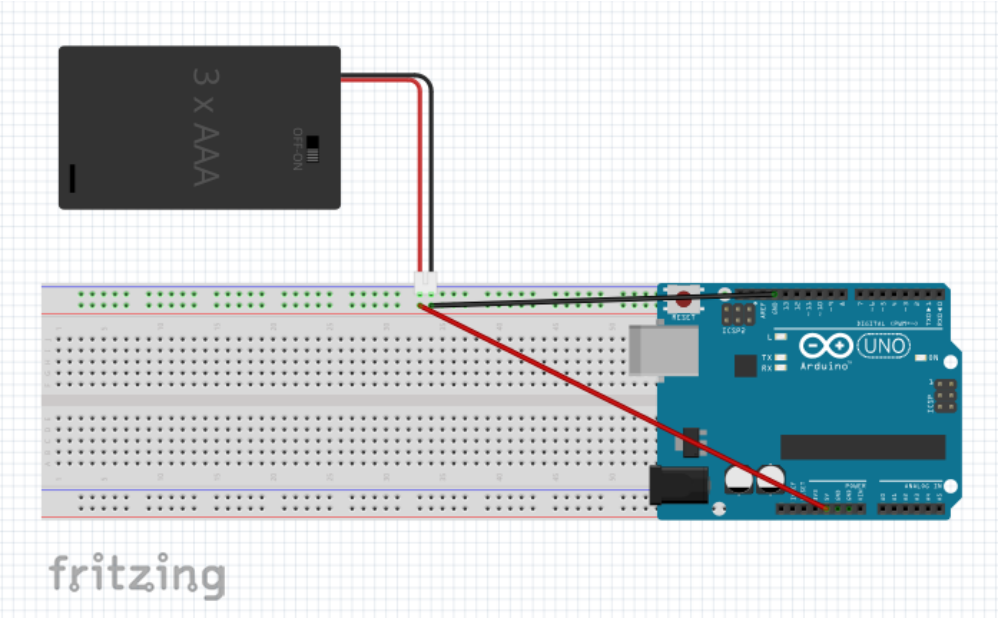

Рисунок 21 – Схема підключення батареї до плати Arduino Piranha UNO

Наступний елемент GSM/GPRS Shield A6 встановлюємо поверх Battery Shield (рис. 22):

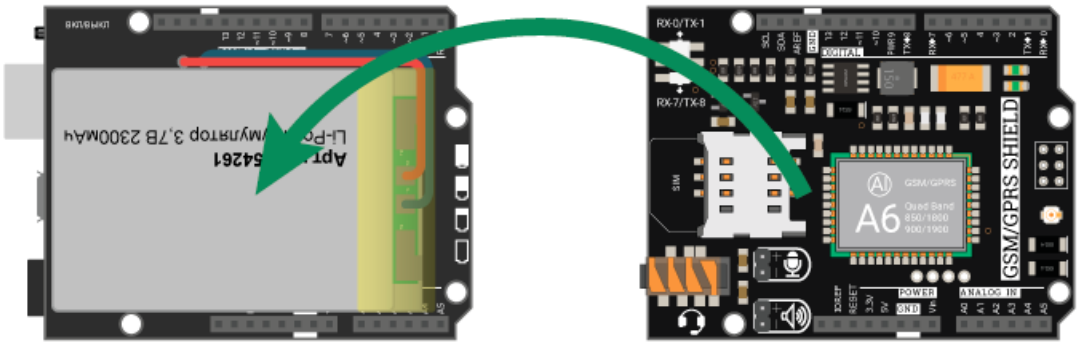

Рисунок 22 – Додавання елементу GSM/GPRS Shield A6 до плати Arduino Piranha UNO

На макетній платі використовується подібний елемент, який аналогічно підключається поверх плати Arduino Piranha UNO це зображено на рис. 23.

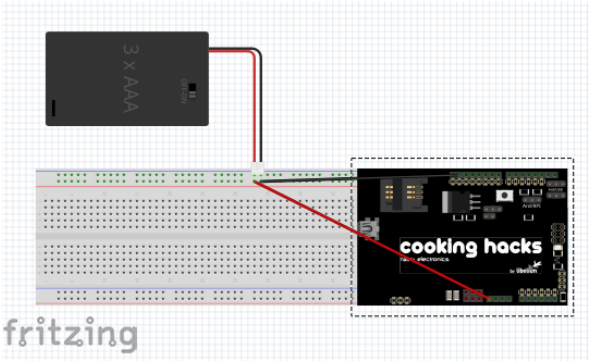

Рисунок 23 – Схема підключення GSM/GPRS Shield до плати Arduino Piranha UNO

Далі для зручності на GSM/GPRS Shield A6 встановлюємо Trema Shield (рис. 24):

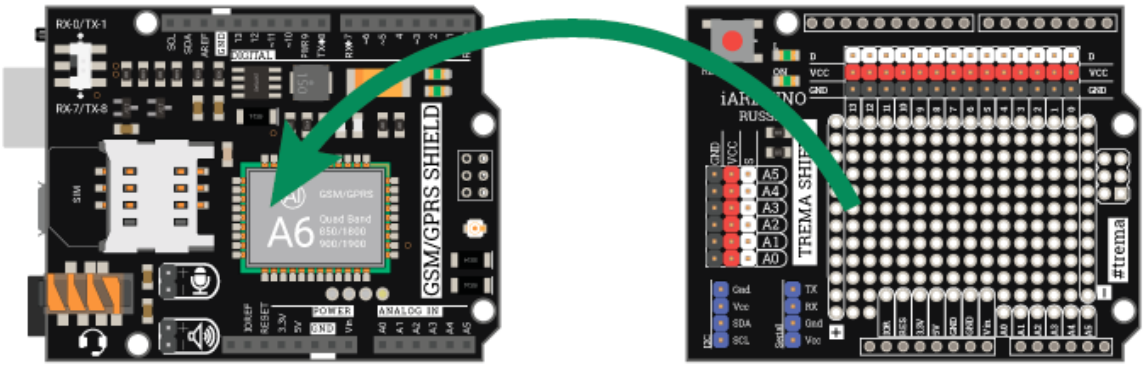

Рисунок 24 – Додавання плати розширення до Arduino Piranha UNO

Підключаємо гарнітуру до GSM/GPRS Shield A6. Також необхідно підключити цифровий термометр до Trema Shield (рис. 25).

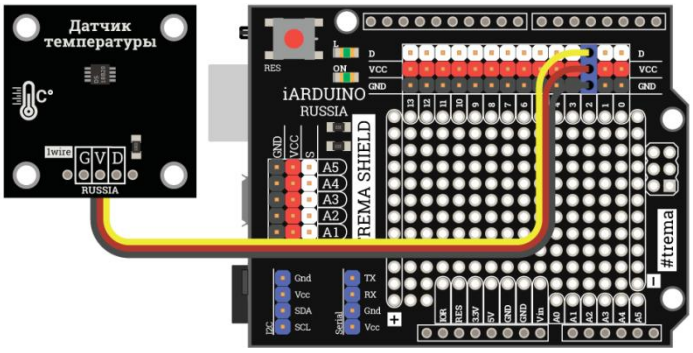

Рисунок 25 – Додавання цифрового термометру до Trema Shield

Наступним етапом буде підключення датчику газу до Trema Shield (рис. 26).

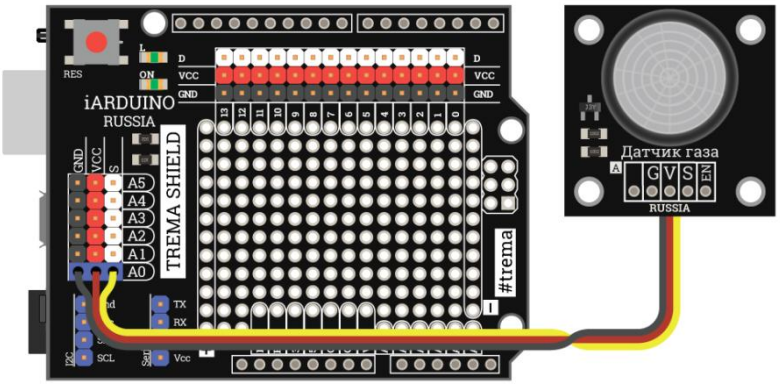

Рисунок 26 – Додавання датчику газу до Trema Shield

Наступним кроком підключаємо цифровий герметичний термометр до Trema Shield (рис. 27). Міжінформаційним проводом (жовтий) і проводом живлення 5В (червоний) встановлено резистор 10кОм.

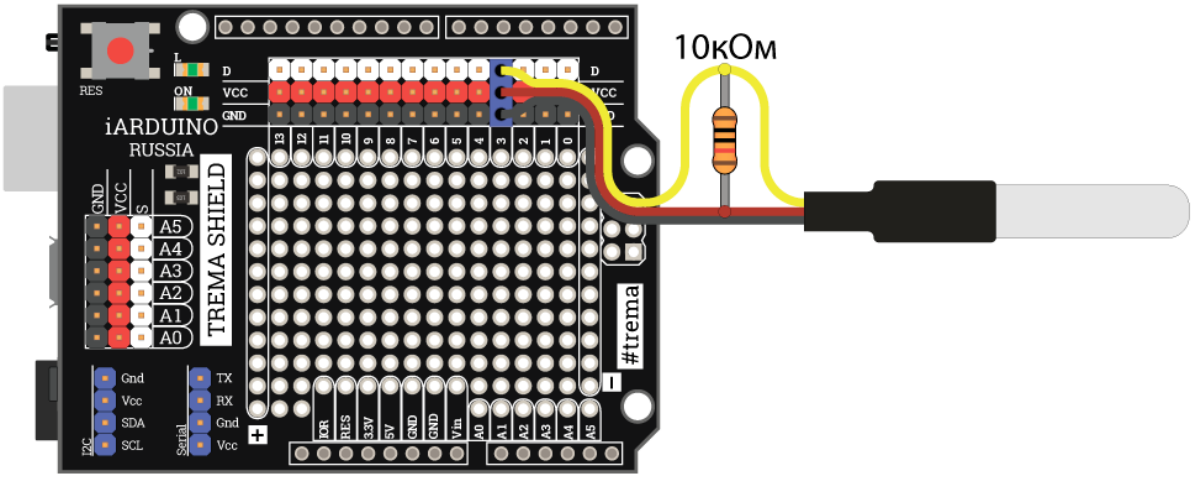

Рисунок 27 – Додавання зовнішнього термометру до Trema Shield

Підключаємо магнітний сповіщувач (геркон) доTremaShield (рис. 28). Від інформаційного дроту (жовтого) через резистор 10кОм підключається дріт GND (чорний).

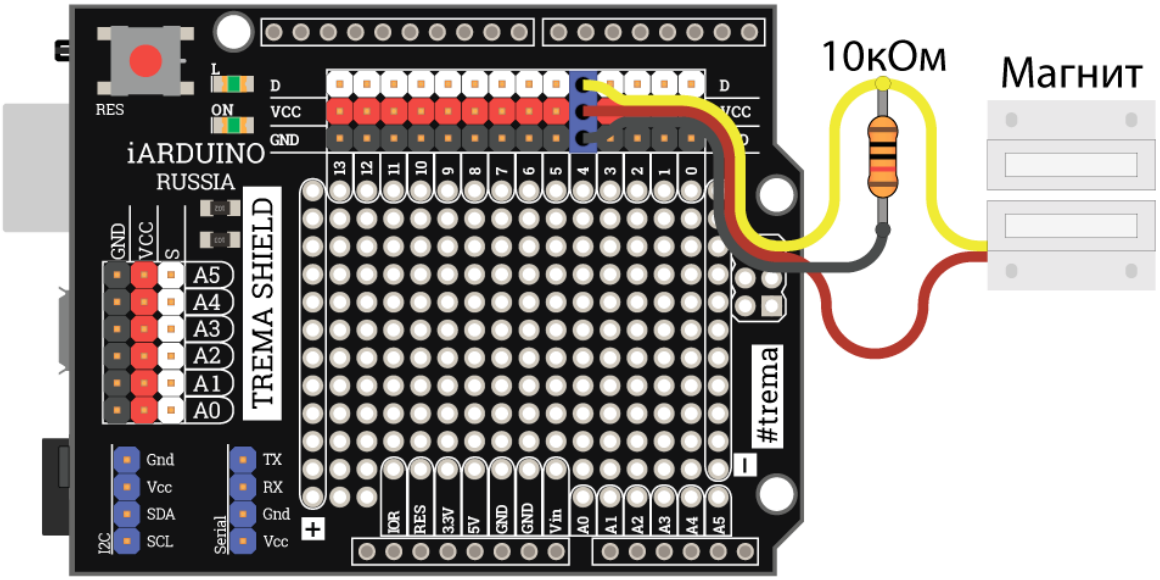

Рисунок 28 – Додавання магнітного сповіщувача до Trema Shield

3 ПРОГРАМУВАННЯ ПРИСТРОЮ

3.1 Опис функціонування системи

Після увімкнення системи необхідно дочекатися СМС повідомлення про те, що ініціалізація пройшла успішно и модем GSM зареєструвався в мережі. Після цього система готова до роботи.

Система може приймати повідомлення з командою "ОТЧЕТ". Після отримання такого повідомлення від користувача система сформую та відправить користувачу повідомлення з даними про стан системи та всіх датчиків.

Також пристрій має кілька сценаріїв роботи з датчиками: при спрацюванні любого з датчиків на телефон користувача відправляється СМС з відповідним повідомленням, яке може містити також інформацію безпосередньо з конкретного датчика.

При спрацюванні деяких особо важливих датчиків система безпеки відправляє СМС з відповідним повідомленням, а після цього здійснює дзвінок на вказаний користувачем номер, завдяки чому абонент може почути, що відбувається на об’єкті, який знаходиться під охороною.

3.2 Опис алгоритму роботи

З початку ми визначаємо та об’являємо змінні та функції, підключаємо бібліотеки, також створюємо об’єкти для подальшої роботи з ними.

Одразу підключаємо бібліотеку для роботи з GSM/GPRS Shield та бібліотеку для програмної реалізації шини UART.

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

Наступні бібліотеки для роботи з OneWire та цифровим термометром DS18B20.

```
#include<OneWire.h>
#include<DallasTemperature.h>
```

Також створюємо об’єкти для подальшої роботи з датчиками температури.

```
OneWireoneWire_in(2);
OneWireoneWire_out(3);
DallasTemperaturesensor_inhouse(&oneWire_in);
DallasTemperaturesensor_outhouse(&oneWire_out);
```

Далі створюємо об’єкт gsm для роботи з функціями і методами бібліотеки iarduino_GSM, а також об’єкт softSerial для роботи по програмній шині UATR

```
iarduino_GSMgsm;
SoftwareSerialsoftSerial(7, 8);
```

Наступним кроком створюємо змінні Textдля зберігання текстової інформації та змінну пошуку символа у рядку.

```
StringText;
StringstrSMStxt;
```

Далі оголошуємо строку для зберігання тексту прийнятих повідомлень,строку для зберігання номеру відправника SMS повідомлень, а також у змінній number задаємо номер, з яким буде працювати система.

```
charSMStxt[161];
charSMSnum[13];
charnumber[13] = "+380978657912";
```

Оголошуємо масив для зберігання покажчика на рядок та Масив з номерами виводів пристроїв

```
char* Func = "ОТЧЕТ";
charPins[] = {A0, 4, 5};
```

Далі вказуємо мінімальну температуру спрацьовування для термометра всередині приміщення:

```
int8_t MIN_TEMP_IN = 15;
```

Максимальну температуру всередині:

```
int8_t MAX_TEMP_IN = 28;
```

Мінімальну та максимальну температуру назовні:

```
int8_t MIN_TEMP_OUT = 10;
int8_t MAX_TEMP_OUT  = 35;
```

Максимальне значення концентрації газу:

```
int16_t MAX_GAS_VALUE = 450;
```

Наступні змінні відповідають за час, після закінчення якого проводиться повторна відправка СМС, час між відправленням СМС та тривожним дзвінком та тривалість дзвінка відповідно.

```
uint32_t delta_time    = 100000;
uint32_t waiting_time  = 5000;
uint32_t dial_waiting  = 30000;
```

Далі вказуємо таймери для тривожного дзвінка, датчики температур, газу, руху, стану на повторне відправлення СМС. За замовчуванням всі вони дорівнюють 0.

```
uint32_t dial_time     = 0;
```

```
uint32_t temp1_time    = 0;
uint32_t temp2_time    = 0;
uint32_t gas_time      = 0;
uint32_t motion_time   = 0;
uint32_t gerkon_time   = 0;
uint32_t call          = 0;
```

Функція voidsetup() містить конфігурацію обраних виводів, ініціалізацію роботи GSM/GPRS Shield. Також вона містить цикл очікування готовності GSM/GPRS Shield до роботи (реєстрації в мережі оператора). Після виконання всіх указаних дій, виконується сповіщення о готовності до роботи, через відправку смс на вказаний номер.

```
void setup() {
sensor_inhouse.begin();
sensor_outhouse.begin();
gsm.begin(softSerial);
```

Чекаємо завершення реєстрації модему в мережі оператора зв'язку.

```
while (gsm.status() != GSM_OK) { delay(1000); }

pinMode(Pins[0], INPUT);
pinMode(Pins[1], INPUT);
```

Установка кодування для символів кирилиці:

```
gsm.TXTsendCodingDetect("п");
```

Та відправляємо повідомлення про вдалий запуск:

```
gsm.SMSsend( F("Система активирована успешно."), number);
}
```

Код void loop() виконується в циклі. При отриманні СМС або при виникненні події, описаної в коді, пристрій відправляє повідомлення абоненту або виконує тривожний дзвінок. Для того щоб після виникнення події модуль

не почав постійно надсилати повідомлення чи робити дзвінки, була додана часова затримка DELTA_TIME, впродовж якої модуль ігнорує повторні події. Після закінчення цього часу, якщо нічого не змінилося, система повторно відправить СМС або здійснить тривожний дзвінок.

```
void loop () {
Text = "";
```

Код програми побудований таким чином, що при видаленні одного з встановлених модулів система залишається повністю працездатною.

Основну частину програми можна розділити на такі блоки:

Датчик газу: у змінній MAX_GAS_VOLUME в самому початку ми задаємо гранично допустимий рівень газу в приміщенні. При досягненні цього значення модуль надійшле на вказаний номер СМС з повідомлення про те, що концентрація газу перевищила норму.

Якщо концентрація газу у нормі, то ці данні будуть внесені до строки Text, яка при запиті користувача буде вислана йому у СМС разом с показниками інших модулей.

```
if (analogRead(Pins[0]) >= MAX_GAS_VALUE) {
if ((gas_time + delta_time) <millis() || gas_time == 0) {
gas_time = millis();
Text = "Превышена концентрация газа.\r\nЗначение = " +
analogRead(Pins[0]);
```

Далі відправляємо повідомлення:

```
gsm.SMSsend(Text, number);
    }
  }
Text = Text + "Газ: " + analogRead(Pins[0]) + "ppm";
delay(10);
```

Датчик температури внутрішній: вказавши в змінних MIN_TEMP_IN і MAX_TEMP_IN границі допустимих температур, налаштуємо модуль на роботу з цифровим термометром, розташованим всередині приміщення.

При досягненні порогового значення модуль відправить на вказаний в змінної number номер СМС з повідомленням про те, що температура змінилася більше, ніж вказано в умові.

Якщо температура всередині приміщення в нормі, то дані будуть внесені в рядок Text, який при запиті користувача буде висланий йому по СМС разом з показниками інших модулів.

```
     sensor_inhouse.requestTemperatures();
     if  (sensor_inhouse.getTempCByIndex(0)  <=  MIN_TEMP_IN  ||
sensor_inhouse.getTempCByIndex(0) >= MAX_TEMP_IN) {
     if ((temp1_time + delta_time) <millis() || temp1_time == 0)
{
           temp1_time = millis();
     Text = "Пороговое изменение температуры внутри.\r\nЗначение
= " + String (sensor_inhouse.getTempCByIndex(0));
     gsm.SMSsend(Text, number);
         }
       }
     Text=Text+"\r\nТвнут:"+sensor_inhouse.getTempCByIndex(0);
     delay(10);
```

Датчик температури зовнішній: вказавши в змінних MIN_TEMP_OUT і MAX_TEMP_OUT границі припустимих температур, налаштуємо модуль на роботу з цифровим термометром, розташованим зовні приміщення.

При досягненні порогового значення модуль відправить на вказаний в змінної number номер смс з повідомленням про те, що температура змінилася більше, ніж вказано в умові.

Якщо температура зовні приміщення в нормі, то дані будуть внесені в рядок Text, який при запиті користувача буде висланий йому по СМС разом з показниками інших модулів.

```
     sensor_outhouse.requestTemperatures();
```

```
     if (sensor_outhouse.getTempCByIndex(0) <= MIN_TEMP_OUT ||
sensor_outhouse.getTempCByIndex(0) >= MAX_TEMP_OUT) {
     if ((temp2_time + delta_time) <millis() || temp2_time == 0)
{
           temp2_time = millis();
     Text = "Пороговое изменение температуры снару-
жи.\r\nЗначение="+String (sensor_outhouse.getTempCByIndex(0));;
     gsm.SMSsend(Text, number);
         }
       }
     Text=Text+"\r\nТвнеш:"+sensor_outhouse.getTempCByIndex(0);
     delay(10);
```

Датчик руху: перевіряємо, чи немає руху в радіусі дії датчика. Якщо рух помічено, то модуль відправляє СМС абоненту на номер, вказаний у змінній number, далі чекає (waiting_time) і після цього робить тривожний дзвінок на той же номер.

В змінної gsm.SOUNDdevice вказано параметр GSM_HEADSET, що означає, що на стороні модуля встановлена гарнітура.

Тривожний дзвінок можна налаштувати:

- модуль здійснює тривожний дзвінок "до переможного", поки абонент не підніме трубку або не спрацює автовідповідач;
- модуль здійснює тривожний дзвінок, а після того, як трубка буде знята, обмежить розмову на час, вказаний в -змінній diall_waiting.

Якщо рух не помічено, то дані будуть внесені до рядка Text, який при запиті користувача буде вислано йому по СМС разом з показниками інших модулів.

```
     if (digitalRead(Pins[1])) {
     if ((motion_time + delta_time) <millis() || motion_time ==
0) {
     motion_time = millis();
     gsm.SMSsend("Замечено движение!", number);
     if ((call + waiting_time) <millis() || call == 0 ) {
```

Чекаємо, поки не пройде час waiting_time, перш ніж зробимо тривожний дзвінок.

```
gsm.SOUNDdevice(GSM_HEADSET);
if (gsm.CALLdial(number)) {
while (gsm.CALLstatus() != GSM_CALL_ACTIVE) {}
```

Чекаємо, поки статус дзвінка не зміниться на GSM_CALL_ACTIVE.

```
dial_time = millis();
while ((dial_time + dial_waiting) >millis() ) {}
```

Чекаємо час dial_waiting, перш ніж автоматично розірвемо з’єднання та розриваємо його.

```
gsm.CALLend();
        }
        //gsm.CALLdial(number);
      }
    }
  }
Text = Text + "\r\nДвижение: -";
delay(10);
```

Датчик стану (геркон): перевіряємо, чи немає розмикання датчика. Якщо датчик розімкнути, то модуль відправляє СМС абоненту на номер, вказаний у змінній number, далі чекає деякий час (waiting_time) і після цього робить тривожний дзвінок на той же номер.

В змінної gsm.SOUNDdevice вказано параметр GSM_HEADSET, що означає, що на стороні модуля встановлена гарнітура.

Тривожний дзвінок можна налаштувати: модуль здійснює тривожний дзвінок "до переможного", поки абонент не підніме трубку або не спрацює автовідповідач.

Модуль здійснює тривожний дзвінок, а після того, як трубка буде знята, обмежить розмову на час, вказаний в змінній diall_waiting.

Якщо датчик замкнутий, то дані будуть внесені в рядок Text, який при запиті користувача буде вислано йому по СМС разом з показниками інших модулів.

```
     if (!digitalRead(Pins[2])) {
     if ((gerkon_time + delta_time) <millis() || gerkon_time ==
0) {
     gerkon_time = millis();
     gsm.SMSsend("Окно открыто!", number);
     if ((call + waiting_time) <millis() || call == 0 ) {
```

Чекаємо, поки не пройде час waiting_time, перш ніж зробимо тривожний дзвінок та робимо його.

```
     gsm.SOUNDdevice(GSM_HEADSET);
     if (gsm.CALLdial(number)) {
     while (gsm.CALLstatus() != GSM_CALL_ACTIVE) {}
     dial_time = millis();
     while ((dial_time + dial_waiting) >millis() ) {}
```

Чекаємо час dial_waiting, перш ніж автоматично розірвемо з'єднання та розриваємо його.

```
     gsm.CALLend();
             }
             //gsm.CALLdial(number);
           }
         }
       }
     Text = Text + "\r\nОкно: +";
     delay(10);
```

Запит звіту: модуль перевіряє раз в секунду, чи немає вхідного повідомлення. Якщо повідомлення є, то модуль перевіряє, чи збігається номер, з якого прийшло повідомлення, з номером, вказаним в змінній number:

Якщо номер збігається, то модуль перевіряє текст повідомлення: якщо в тексті повідомлення вказана команда з змінної Func, то модуль відправляє у відповідь СМС з повним звітом по всіх датчиках.

Якщо в тексті СМС помилка, то модуль у відповідному СМС повідомить про це. У разі, якщо номер не збігається, то модуль буде ігнорувати текст СМС:

```
if (millis() % 1000 < 100) {
delay(100);
if (gsm.SMSavailable()) {
gsm.SMSread(SMStxt, SMSnum);
if (!strcmp(number, SMSnum)) {
strSMStxt = SMStxt;
if (strSMStxt.indexOf(Func) != -1 ) {
gsm.SMSsend(Text, number);
        } else {
gsm.SMSsend(F("Ошибка в тексте!"), number);
        }
      }
    }
  }
```

Відповідь на вхідний дзвінок: модуль так само підтримує голосові вхідні дзвінки для того, щоб абонент зміг у будь-який час дізнатися, що відбувається у приміщенні, яке знаходиться під охороною.

В змінної gsm.SOUNDdevice вказано параметр GSM_HEADSET, що означає, що на стороні модуля встановлена гарнітура.

Для того, щоб модуль відповів на вхідний дзвінок, номер, з якого здійснюють виклик, повинен збігатися з номером, вказаним в змінної number.

```
if (gsm.CALLavailable(SMSnum)) {
if (!strcmp(number, SMSnum)) {
gsm.SOUNDdevice(GSM_HEADSET);
gsm.CALLup();
while (gsm.CALLavailable()) {}
    }
  }

}
```

ВИСНОВКИ

На даний момент іде активна розробка автоматизованих систем безпеки будинків. Незважаючи на це, процеси стандартизації та глобалізації почалися порівняно нещодавно й на даний час існує безліч різноманітних рішень.

Незважаючи на певний перелік недоліків, такі технології як «Розумний будинок» та різні системи безпеки помешкань з високою вірогідністю займуть своє місце у сфері побутової електроніки і перейдуть з розряду рідкостей в повсякденність. Цьому сприяють і появи простих і функціональних контролерів сімейства Arduino та підвищення обчислювальної потужності мобільних пристроїв. Однак поки такі технології для багатьох так і залишаються занадто дорогими. Незважаючи на позитивні тенденції в сфері, залишилася потреба в недорогому і вільному варіанті реалізації, який кожен зможе налаштувати під свої потреби не володіючи технічними навичками і який зможе працювати з більшістю вже створених пристроїв і контролерів.

У даній дипломній роботі було спроектовано розумну систему безпеки будинку на базі мікроконтролеру Arduino, а також були розглянуті основні види систем безпеки, компоненти розумної системи безпеки, середовище проектування мікроконтролерів Arduino, а також було проаналізовано декілька популярних аналогів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Системы безопасности загородного дома, виды охранных систем. URL: https://bezopasnostin.ru/ohrannaya-signalizatsiya/bezopasnost-doma.html. (дата звернення 12.03.2019).
2. Система ограниченного доступа вашего дома. URL: https://sovtehnika.ru/sistema-ogranichennogo-dostupa-vashego-doma.html. (дата звернення 12.03.2019).
3. Система контролю доступу. URL: https://leater.com/ua/services/sistema-kontrolyu-dostupu.html. (дата звернення 13.03.2019).
4. Відеонагляд. URL: http://tdim.com.ua/ua/systemy-bezpeky/videonahliad.html. (дата звернення 14.03.2019).
5. Скрытое видео и аудіо наблюдение. URL: https://guard-lviv.com.ua/info/skrytoe-video-i-audio-nablyudenie. (дата звернення 14.03.2019).
6. Автономные охранные сигнализации. URL: https://www.forter.com.ua/news-and-articles/avtonomnye-ohrannye-signalizatsii-kak-eto-rabotaet/. (дата звернення 14.03.2019).
7. Розумний будинок. URL: https://www.smarthouse.ua/ua/umnyj_dom.html. (дата звернення 14.03.2019).
8. Ajax Systems. URL: https://ajax.systems/. (дата звернення 14.03.2019).
9. Pitbull Alarm Intrusion Alarm Panel, Eldes Alarms. URL: https://eldesalarms.com/product/pitbull-alarm/. (дата звернення 15.03.2019).
10. Умная сигнализация Jablotron. URL: https://www.jablotron.com/ru/. (дата звернення 15.03.2019).

11. Сигнализация в дом. Развитие технологий и тренды 2018. URL: https://krok-ua.com/signalizacija-dlja-doma-i-kvartiry-novye-trendy-razvitija-v-2018-godu/. (дата звернення 15.03.2019).
12. Среда разработки Arduino. URL: http://arduino.ru/Arduino_environment. (дата звернення 15.03.2019).
13. Arduino ТОП-3 графических сред программирования CHIP. URL: https://ichip.ru/arduino-top-3-graficheskikh-sred-programmirovaniya.html. (дата звернення 17.03.2019).
14. Основы UML – диаграммы использования (use-case). URL: https://pro-prof.com/archives/2594. (дата звернення 23.04.2019).
15. UML UseCaseDiagramTutorial. URL: https://www.lucidchart.com/pages/uml-use-case-diagram. (дата звернення 23.04.2019).
16. Моделирование на UML. URL: http://book.uml3.ru/. (дата звернення 26.04.2019).