Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота
магістра

з теми: «**РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ЕЛЕМЕНТАМИ IOT НА ОСНОВІ OPENHAB**»

Виконав: студент групи KN1-M21,
спеціальності 122 Комп'ютерні науки
Дідик Богдан Петрович

Керівник: **Понеділок В. В.,**
кандидат технічних наук, старший
викладач кафедри комп'ютерних наук

Рецензент: **Оптасюк С. В.,**
кандидат фізико-математичних наук,
доцент, завідувач кафедри фізики

Кам'янець-Подільський – 2022

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1. ПОНЯТТЯ ІНТЕРНЕТУ РЕЧЕЙ 4
1.1. Інтернет речей 4
1.2. Застосування Інтернету речей 4
1.3. Екосистема Інтернету речей 7
1.4. Архітектура Інтернету речей 9
1.5. Розумний будинок 12
1.6. Основні функції розумного будинку 13
1.7. Вимоги до розумного будинку 14
РОЗДІЛ 2. ОГЛЯД ПЛАТФОРМИ OPENHAB ТА МІКРОКОНТРОЛЛЕРА ARDUINO 17
2.1. Платформа OpenHAB 17
2.2. Порівняння OpenHAB та Home Assistant 18
2.3. Встановлення та налаштування OpenHAB 21
2.4. Мікроконтроллер Arduino 24
РОЗДІЛ 3. ПРОЕКТУВАННЯ СИГНАЛІЗАЦІЇ ТА НАЛАШТУВАННЯ УПРАВЛІННЯ НЕЮ З ДОПОМОГОЮ OPENHAB 27
3.1. Встановлення MQTT брокера 27
3.2. Налаштування зв’язку OpenHAB з MQTT брокером 29
3.3. Налаштування розумного пристрою в OpenHAB 32
3.4. Створення каналу взаємодії зі станом сигналізації 33
3.5. Налаштування інтерфейсу для керування станом 34
3.6. Проектування схеми та написання коду в Arduino IDE 37
ВИСНОВКИ 42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43
ДОДАТКИ 44

ВСТУП

Актуальність теми. Сьогодні глобальна інформатизація суспільства і активний процес науково-технічного розвитку в області інформаційних систем сприяють формуванню єдиного світового інформаційного простору. Однією з перспективних тенденцій розвитку сучасних інформаційних систем та технологій стає розширення доступності інформаційно-обчислювальних ресурсів мереж для окремих абонентів, в тому числі і речей. Сьогодні пристрої Інтернету речей не лише масово використовуються у щоденному вжитку, але й у сучасному бізнес-середовищі. Зокрема Інтернет речей активно впроваджується в різноманітних галузях — від машинобудування та систем телекомунікацій до сільського господарства і будівництва. Поступово пристрої Інтернету речей стають невід'ємною частиною людського життя, і зростання їх кількості спричиняє попит на їх освоєння та впровадження для виконання різноманітних задач.

Метою роботи є огляд сучасного стану Інтернету речей та з врахуванням особливостей здійснити практичну реалізацію телекомунікаційної взаємодії мікроконтроллера Arduino та програмного комплексу OpenHAB.

Відповідно до поставленої мети були сформульовані наступні **завдання дослідження:**

1) Провести огляд Інтернету речей.
2) Розглянути програмний комплекс OpenHAB та мікроконтроллер Arduino.
3) Реалізувати керування платою Arduino, використовуючи OpenHAB.

Об'єктом дослідження є реалізація взаємодії елементів IoT з програмним комплексом OpenHAB.

Предметом дослідження є програмний комплекс OpenHAB зі всіма його інтеграційними можливостями та мікроконтроллер Arduino.

Структура роботи. Дана магістерська робота складається зі вступу, теоретичної частини, практичної частини, висновків та додатків.

РОЗДІЛ 1. ПОНЯТТЯ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Інтернет речей

Інтернет речей (англ. Internet Of Things, IoT) — це всесвітня мережа пристроїв, що з'єднані між собою та обмінюються між собою інформацією без людського втручання. Такі пристрої використовують мережу Інтернет для транслювання основних даних у хмару, звідки інші пристрої збирають ці відомості для вирішення різноманітних задач. Раніше ця технологія використовувалася в межах проектування "розумного будинку" для керування освітленням, сигналізацією, температурою тощо у приміщенні. Зараз IoT застосовують усі галузі, де необхідна автоматизація процесів.

Термін «Інтернет речей» запропонував Кевін Ештон у 1999 році. Він висловив припущення, що незабаром у кожної з речей реального фізичного світу в IoT буде цифровий двійник, її віртуальне представлення. Напрямок IoT почав активно розвиватися, починаючи з 2000-х років, коли кількість користувачів Інтернету була перевищена кількістю пристроїв, підключених до мережі Інтернет [1].

Виходячи з даних швецької компанії Ericsson, сьогодні у світі налічується більше ніж 16 мільярдів підключених до Інтернету пристроїв. Уже станом на 2018 рік їх кількість перевищила сумарну кількість мобільних телефонів у світі. Прогнозується, що 2030 року число IoT-пристроїв буде сягати майже 30 мільярдів [2].

1.2. Застосування Інтернету речей

Сьогодні Інтернет речей вже наявний в багатьох сферах нашого життя (рис. 1.1) [3]. Ось приклади:

- ☑ **Розумний будинок.** На сьогоднішній день розумний будинок є однією х найпопулярніших сфер застосування Інтернету речей. Існує безліч пристроїв, які можуть виконувати корисні дії по дому. Наприклад, існують розумні чайники, пилососи, колонки, навіть

годівниці для тварин та інші повсякденні пристрої, які виконують звичайні домашні функції.

- ☑ **Розумні автомобілі.** Сьогодні безпілотні машини вже здатні витіснити звичайні авто. Такі автомобілі детально прораховують маршрут до пункту призначення. Застосування штучного інтелекту разом з купою датчиків здатні забезпечити комфорт та безпеку.
- ☑ **Охорона здоров'я**. Сфера охорони здоров'я займає одну з найважливіших ланок в переліку сфер застосування Інтернету речей. Вплив на життя людей показує важливість медицини, як сфери діяльності в сучасному суспільстві. В генетиці завдяки IoT робляться відкриття, а в індивідуальному методі лікуванні Інтернет речей дозволяє знайти підхід до кожного пацієнта окремо, аналізуючи стан його здоров'я.
- ☑ **Агрокультура**. Інтернет речей активно використовується в сільському господарстві: у фермерстві, сфері розведення тварин, сфері вирощування урожаю тощо. Також IoT-пристрої в цій сфері дають змогу надавати перевірку складу ґрунту, прогнозувати кліматичні зміни, відстежувати стан здоров'я хворих тварин, автоматично поливати посівних культур, знешкоджувати нездорові області урожаю тощо.
- ☑ **Промисловість**. В цій сфері застосовуються різноманітні сенсори, програмні системи та аналіз великих даних для розробки дизайнів та точних підрахунків. Розумна спеціалізована техніка здатна покращити продуктивність та виправляють людські помилки, які пов'язані із контролем якості.
- ☑ **Аксесуари**. До цієї сфери відносяться пристрої можна, якими можна керувати за допомогою додатку на смартфоні та які носяться людиною на тілі. Такі пристрої пов'язані з медичними IoT-пристроями, оскільки з їх допомогою можна відслідковувати

базові медичні показники здоров'я та покращувати їх за допомогою певної стратегії лікування. Наприклад, такі компанії як Apple, Samsung та Motorola розробляють розумні фітнес-браслети, розумні імплантати та інші пристрої.

- ☑ **Ритейл.** В цій сфері завдяки IoT-пристроям найбажаніші товари та послуги з'являються прямо перед очима клієнта в потрібний момент. Інтернет речей дозволяє точніше налаштувати рекламу, покращити процес постачання та процес аналізу найпопулярніших товарів. Якщо говорити про Інтернет речей в торгівлі, то сюди відноситься безконтактна оплата та спеціальні додатки для здійснення покупок через мережу Інтернет.
- ☑ **Розумне місто**. До цієї сфери відноситься розумне паркування, карти шуму, розумне освітлення та дороги. Зараз ця сукупність має великі перспективи в проектуванні міст, хоча зараз все знаходиться на стадії планування та розробки. Також завдяки IoT-технологіям можна збільшити безпеку на міських дорогах, краще контролювати рух міського транспорту та забруднення великих індустріальних міст.
- ☑ **Канал постачання**. У цій сфері Інтернет речей дозволяє відстежувати товари, слідкувати за станом доставки та вести відкритий обмін інформацією між ключовими учасниками процесу постачання. Інтернет речей зменшує кількість робочих місць, що веде до зниження витрат та покращенню автоматизації праці.
- ☑ **Енергетика**. Завдяки IoT розумна електромережа може автоматично збирати необхідні дані та моментально аналізувати циркуляцію напруги. Як наслідок, клієнти та постачальники зможуть оптимізувати використання електрики.

Рис 1.1 Застосування Інтернету речей

1.3. Екосистема Інтернету речей

Екосистема Інтернету речей (рис. 1.2) складається з усіх засобів, сервісів та технологій, які використовуються в Інтернеті речей [4].

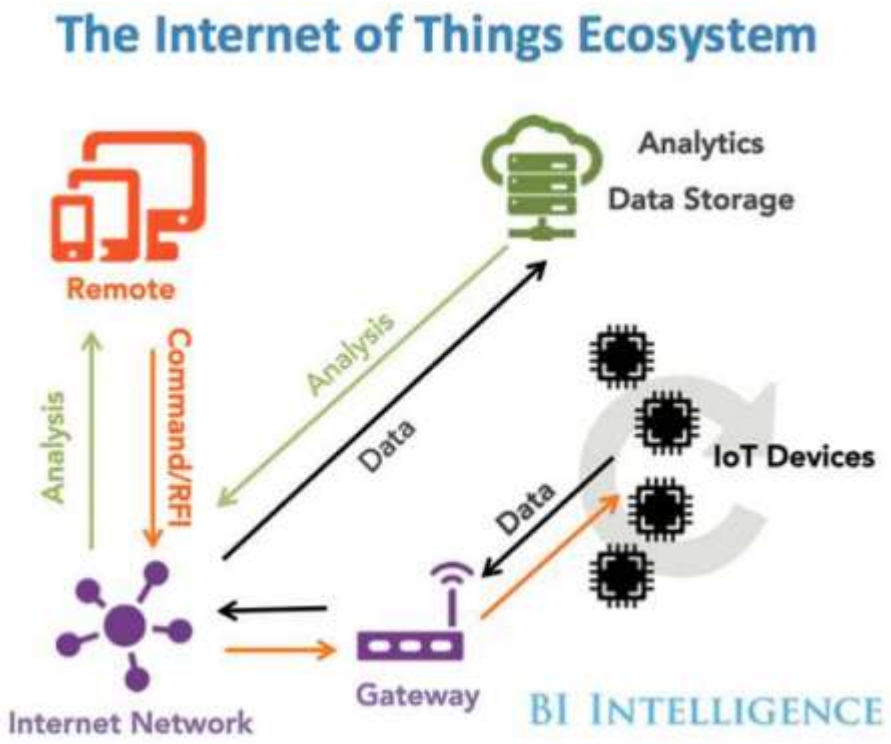

Рис 1.2 Екосистема Інтернету речей

До екосистеми можна віднести:

- розумні датчики / виконавчі механізми — вбудовані системи, операційні системи, що працюють в режимі реального часу, джерела безперебійного живлення, мікро-електромеханічні системи;
- системи зв'язку з датчиками — системи для обміну даними між датчиками, що застосовують низькошвидкісні малопотужні інформаційні канали, які часто побудовані не на протоколі IP;
- локальні обчислювальні мережі (LAN) — системи обміну даними, що використовують протокол IP;
- маршрутизатори (routers), шлюзи (gateways), пограничні пристрої (edge device);
- глобальна обчислювальна мережа (WAN);
- хмара — інфраструктура, що є постачальником послуг: інструментів для аналізу даних, програмного забезпечення, сервісів машинного навчання тощо;

- сервіси аналізу даних — величезні масиви інформації, що передаються в хмару. Робота з великими обсягами даних і отримання з них користі вимагає комплексної аналітики та застосування прийомів машинного навчання;
- безпека — стосується кожного компонента: від датчиків до апаратного забезпечення, систем радіозв'язку і самих протоколів передачі даних. Кожен рівень необхідний мати безпеку, достовірність і цілісність. В ланцюзі не повинно бути слабких місць, оскільки Інтернет речей стане мішенню для атак хакерів.

1.4. Архітектура Інтернету речей

На сьогодні існує безліч видів архітектурних рішень, основного поки що не існує, оскільки перші спроби стандартизації області були зроблені недавно. Відсутність загальноприйнятих стандартів призвела до появи різноманітних розрізнених систем, таких як «розумний будинок» (smart home), від різних компаній, що призводить до наступних проблем:

- створення великої кількості стандартів;
- надмірне регламентування найбільш простих об’єктів і процесів;
- збиток загальних завдань стандартизації внаслідок лобіювання інтересів окремих компаній;
- довгі терміни розробки стандартів призводять до їхнього морального старіння, оскільки вони не встигають за розвитком суміжних технологій.

Наразі виділяють дві основні архітектури: програмну і фізичну. Програмна архітектура (рис. 1.3) схожа до моделі OSI та складається з 7 рівнів [5]:

1. Фізичний рівень. Сюди відносяться датчики та електронні пристрої, які здатні підключатися до хмари і отримувати дані від неї.

2. Канальний рівень. Датчики збирають дані, але їх потрібно перетворити в зрозумілий формат і підключити цей пристрій до системи, використовуючи протокол обміну даними, який потрібно налаштувати.
3. Підключення до мережі. Підключення пристрою до провідної або бездротової мережі.
4. Акумуляція даних. Цей рівень відповідає за рівнь безпеку та доступ до даних. Цей рівень має бути «мобільним» для внесення швидких змін.
5. Абстрагування даних. Цей рівень використовує зібрані дані для прийняття рішення або для цілей звітності. Це важливий рівень, в який входить створене рішення і бізнес-логіка.
6. Рівень додатків. На цьому рівні відбувається контроль, аналіз та подання звітів системи. Використовуючи ці дані, можна відображати звіти або застосовувати машинне навчання чи якусь спеціальну логіку або використовувати інтелектуальне рішення і посилати сигнал назад на датчики.
7. Прикладний рівень. Цей рівень призначений для інтерфейсу користувача.

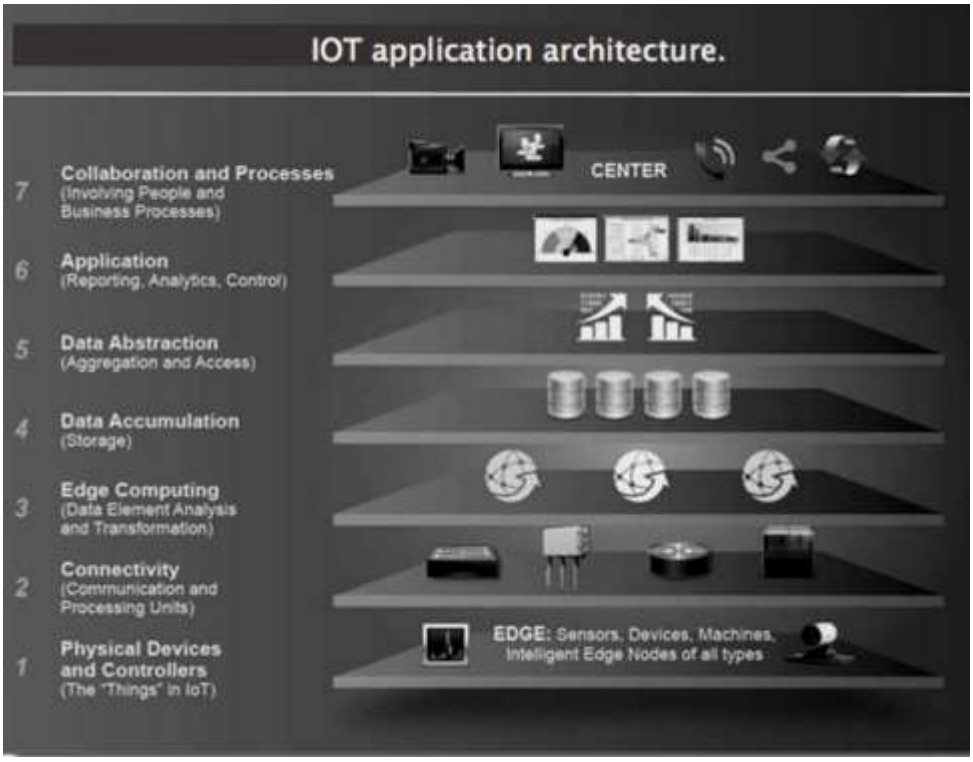

Рис 1.3 Програмна архітектура Інтернету речей

Фізична архітектура (рис. 1.4) Інтернету речей являє собою взаємодію всіх учасників мережі (датчиків, пристроїв), а також обробку отриманої інформації та прийняття рішень.

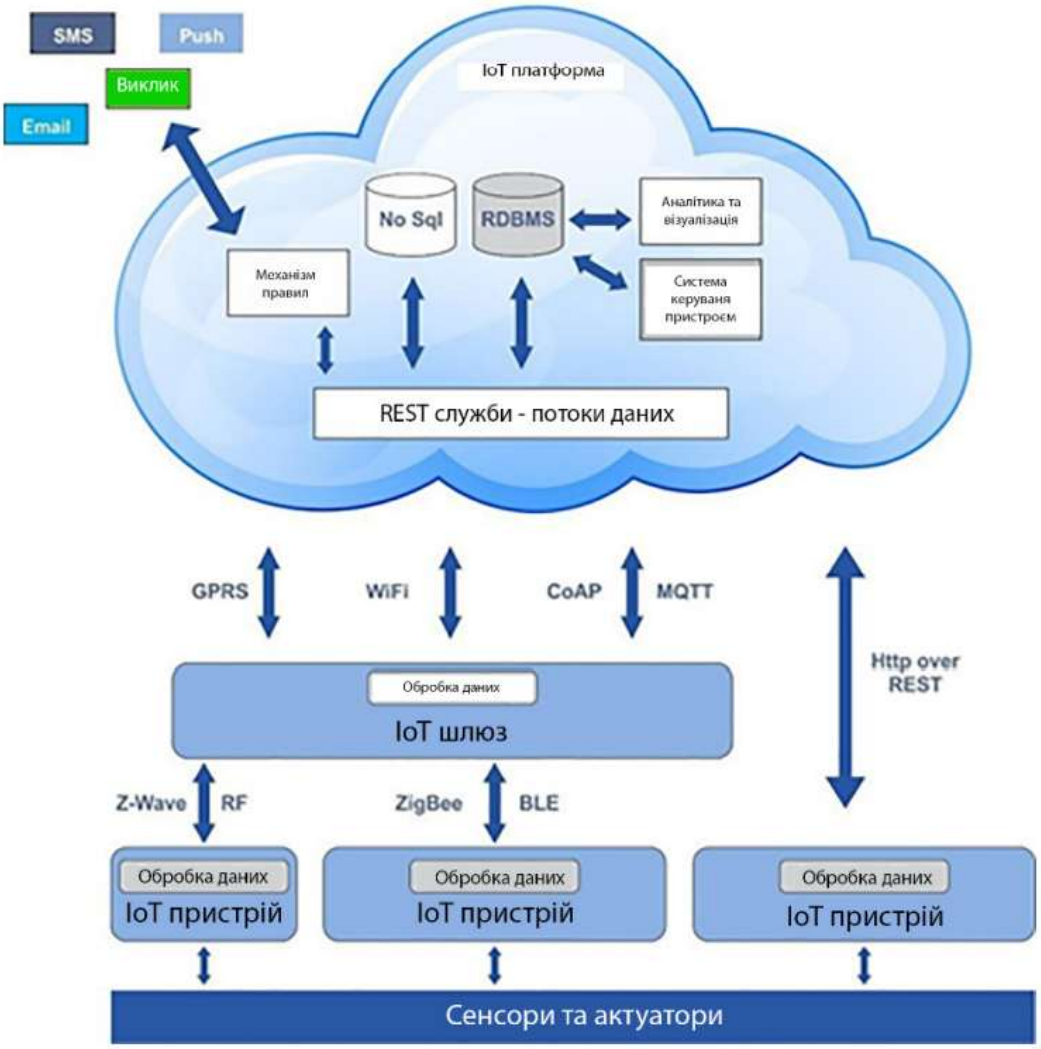

Рис 1.4 Фізична архітектура Інтернету речей

1.5. Розумний будинок

Розумний будинок (розумний дім, smart home, digital house) — будинок, дача чи приміщення комерційного призначення (бутік, офіс, будь-яка установа), які мають системи забезпечення та операційний multi-room. Мульти-рум означає, що усі електроприлади будинку функціонально пов'язуються між собою для централізованого керування. IoT-пристрої можуть бути під'єднані до комп'ютерної мережі, що дозволяє керувати ними за допомогою комп'ютера та надає віддалений доступ до них через мережу Інтернет. Інтеграції інформаційних технологій у домашні умови дозволяє усім пристроям узгоджувати виконання функцій між собою, порівнюючи задані зовнішні показники.

Розумний дім створюється різноманітними компаніями з використанням професійного проектування. Проектування забезпечує програми, що розраховані на певні потреби мешканців та ситуації, пов'язані із зміною середовища або безпекою. Особливість розумного будинку — керування розумними пристроями з пульта дистанційного управління. Система мульти-рум сама аналізує навколишню ситуацію, виходячи з параметрів приміщення, та, керуючись власними висновками, виконує поставлені користувачем команди із відповідними налаштуваннями [6].

Розумний будинок складається з сукупності різноманітних модулів, кожен з яких має своє застосування і підходить для конкретного випадку. Їх можна розділити на 5 підгруп:

- логічні пристрої;
- керуючі пристрої;
- керовані пристрої;
- датчики;
- шлюзи зв'язку;

1.6. Основні функції розумного будинку

Основні функції розумного будинку наступні [6]:

- керування здійснюється з допомогою пульта-дисплея;
- надійна та проста охоронна система та система відео-нагляду;
- контроль за протіканням води/газу;
- можливість керування розумними компонентами будинку та побутовими приладами з допомогою мережі Інтернет. Це означає, що можна робити домашні справи через смартфон або веб-браузер, при цьому фізично не перебуваючи в будинку;
- автоматична централізована корекція освітлення в залежності від часу та перебування людей в конкретному приміщенні (актуально при вихованні дітей або догляду за родичами похилого віку);
- наявність енерговитратної системи розумного будинку;

- наявність облаштуваного приміщення згідно із вихованням дитини: її розвиток, безпека та розваги;
- облаштування приміщень для різних видів ігор;
- наявність домашньої автоматики, що покращує умови життя та спрощує побутові задачі для користування людьми похилого віку чи людьми з обмеженими можливостями;
- виконання побутових турбот за допомогою сучасного обладнання. Це, наприклад, може бути полив квітів або їх накриття від сонця, опираючись на виміри погодних умов, відчинення дверцят о певній годині для вигулу домашніх тварин.

1.7. Вимоги до розумного будинку

Сьогодні зростає кількість розумних пристроїв в будинку, потрібен універсальний спосіб, за допомогою якого вони будуть спілкуватися між собою. Для цього існує кілька несумісних між собою бездротових стандартів передачі сигналу. До цих стандартів відносяться не лише Wi-Fi та Bluetooth, а й спеціалізовані протоколи – Z-Wave, ZigBee та Thread (рис. 1.5). Ці протоколи мають переваги та недоліки які є не очевидними з першого погляду. Вибір передачі сигналу між розумними пристроями потрібно зробити відтоді, як з'явиться перший пристрій розумного будинку [7].

Рис 1.5 Бездротові стандарти зв'язку для розумного будинку

Загальні вимоги до бездротових технологій домашньої автоматизації:

1) **Енергоспоживання**. Ідеальний пристрій розумного будинку повинен використовувати бездротовий приймач та передавач. Обидва мають споживати мінімальну кількість енергії для того, щоб пристрій міг працювати протягом довгого часу без необхідності заміни батареї. Тому вимоги до енергоспоживання пристроїв розумного будинку повинні бути дуже жорсткими.
2) **Радіус дії та безпека**. Сигнал від розумного пристрою повинен стабільно та з мінімальною затримкою досягати іншого розумного пристрою, який розміщений на іншій віддаленій ділянці житла. Саме тому сигнали від будь-якого пристрою повинні долати всі перешкоди, в тому числі поширюватися через стіни та підлоги будинку. Всі IoT-пристрої, незалежно від їхньої кількості, повинні працювати злагоджено, та зводити до мінімуму перешкоди від інших бездротових мереж, шо працюють на тій же частоті. Також потрібно не забувати і за безпеку: повідомлення пристрою повинні бути захищені шифруванням, а процедура додавання нового пристрою не повинна ускладнювати безпеку всієї системи.
3) **Робота за розкладом**. Деякий функціонал розумних пристроїв дозволяє встановити розклад для таких подій, як, наприклад, включення світла, включення опалення, відкриття жалюзів тощо. Проте є багато подій в будинку які неможливо передбачити заздалегідь. Серед таких подій: витік газу, протікання води, поява диму тощо. Тому важливо, щоб кожен розумний пристрій в межах будинку міг відправити або отримати команду на виконання тих чи інших дій, які сталися не за сценарієм.
4) **Відмовостійкість**. Такий фактор як стійкість до відмови залежить від топології бездротової мережі. Також топологія прямо впливає на енергозбереження та радіус дії пристрою. Сучасний підхід для

побудови бездротових мереж передбачає децентралізований підхід з використанням пористої топології (mesh-мережа). Кожен пристрій у такій мережі безпосередньо зв'язується з іншим пристроєм у радіусі своєї дії. При умові, якщо два пристрої знаходяться далеко один від одного, то сигнали будуть передаватися через проміжні пристрої цієї мережі, збільшуючи зону дії пристроїв.

5) **Взаємна сумісність**. Велика кількість розумних пристроїв в будинку означає що всі вони мають спілкуватися між собою. Проте потрібно, щоб пристрої від різних виробників могли гарантувати безперешкодне спілкування та гарантувати взаємну сумісність.

РОЗДІЛ 2. ОГЛЯД ПЛАТФОРМИ OPENHAB ТА МІКРОКОНТРОЛЛЕРА ARDUINO

2.1. Платформа OpenHAB

OpenHAB (Open Home Automation Bus) — це платформа домашньої автоматизації з відкритим вихідним кодом, яка не залежить від технологій і працює як центральна система розумного будинку [8].

Сильні сторони OpenHAB:

- Здатність інтегрувати безліч інших пристроїв і систем. OpenHAB об’єднує інші системи домашньої автоматизації, (розумні) пристрої та інші технології в одне рішення.
- Забезпечення єдиного інтерфейсу користувача та загальний підхід до правил автоматизації для всієї системи, незалежно від кількості залучених виробників і підсистем.
- Наявність найгнучкішого доступного інструменту для втілення майже будь-якого рішення домашньої автоматизації.

OpenHAB спілкується з розумними пристроями, виконує визначені користувачем дії та надає веб-сторінки з визначеною користувачем інформацією, а також визначені користувачем інструменти для взаємодії з усіма пристроями. Щоб досягти цього, OpenHAB сегментує та розподіляє певні функції та операції. загальний опис найважливіших понять OpenHAB (рис. 2.1).

- Прив’язки (Bindings) — це компонент OpenHAB, який надає інтерфейс для електронної взаємодії з пристроями.
- Речі (Things) — це представлення пристроїв.
- Канали (Channels) — це зв’язок між речами та предметами.
- Елементи (Items) — це представлення інформації про пристрої.
- Правила (Rules) — які виконують автоматичні дії

- Інтерфейс (Sitemap) — це створений інтерфейс користувача (веб-сайт), який представляє інформацію та дозволяє взаємодіяти з елементами.

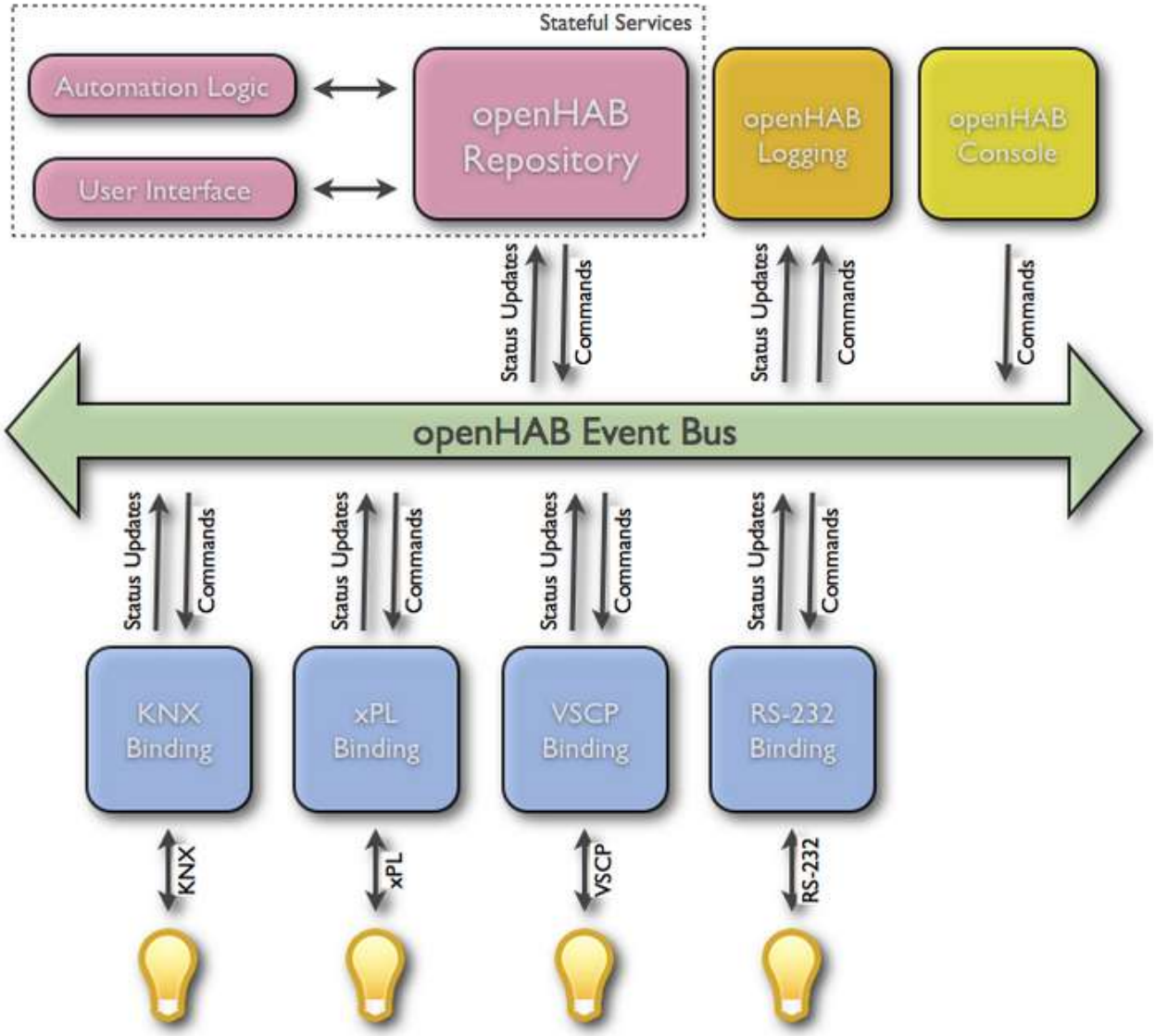

Рис. 2.1 Архітектура OpenHAB

2.2. Порівняння OpenHAB та Home Assistant

На даний момент серед лідерів в сфері домашньої автоматизації наявні дві платформи — OpenHAB та Home Assistant.

Система домашньої автоматизації OpenHAB надає користувачам додаткові модулі, які надають платформі різноманітні можливості та можливість взаємодії [9].

Переваги:

- ☑ Підтримка тисячі пристроїв, численних систем та технологій.
- ☑ Працює на таких платформах, як Linux, Mac OSx, Windows, PINE64, Raspberry Pi, Docker, Synology та інших платформах.
- ☑ Має додаткові панелі Lovelace, програмні компоненти та багато іншого.
- ☑ Працює з Zigbee і Z-Wave.
- ☑ Широка спільнота.
- ☑ Чудова документація.
- ☑ Зручне використання наступних надбудов:
  - Дії: Telegram, Twitter, MQTT, XMPP
  - Прив'язки: ZigBee, Astro, IKEA
  - Зберігання даних: MySQL, InfluxDB, MongoDB
  - Перетворення даних: JavaScript, Map, Exec, XPath
  - Системна інтеграція: HomeKit, Google, OpenHAB cloud connector
  - Читання тексту вголос: Google Cloud Text-to-speech

Недоліки:

- ☒ Ширший функціонал означає більшу трату часу на освоєння.
- ☒ Для безпроблемного використання потрібні знання.
- ☒ Оновлення потребують інтерфейсу командного рядка (CLI).
- ☒ Старіші версії, як-от Raspberry Pi, можуть викликати проблеми через низький обсяг оперативної пам’яті.
- ☒ За тиждень публікується менше тем порівняно з Home Assistant.

Home Assistant — це подібний до OpenHAB центр домашньої автоматизації з відкритим вихідним кодом, розроблений з використанням Python як базової мови..

Home Assistant є більш легким варіантом порівняно з OpenHAB, але він надає величезний набір новітніх функцій. Конфігурація в основному

виконується через веб-інтерфейс, а плагіни можна використовувати для розширення функцій.

Переваги:

- ☑ Привабливий інтерфейс користувача.
- ☑ Підтримує як Google Assistant, так і Alexa від Amazon.
- ☑ Відкритий код, який доступний на GitHub.
- ☑ Гнучкі правила автоматизації, які користувачі можуть налаштувати за своїм бажанням.
- ☑ Функція «інтеграції» у веб-інтерфейсі, значно економить час.
- ☑ Покрокова інструкція на сайті для нових користувачів.
- ☑ Більш стабільні повідомлення та оповіщення.
- ☑ Активна спільнота, яка щотижня публікує близько 500 тем у 10 різних категоріях.
- ☑ Зручне використання наступних надбудов:
  - Almond
  - CEC Scanner
  - Configurator
  - deConz
  - DHCP Server
  - Git pull

Недоліки:

- ☒ Є деякі помилки у верхньому та нижньому регістрах, які можуть спричинити проблеми.
- ☒ Використання YAML може бути складним для деяких користувачів.
- ☒ Новим користувачам спочатку це може здатися складним для освоєння.

OpenHAB, безперечно, є одним із лідерів у сфері Інтернету речей, однак у порівнянні з Home Assistant темпи розробки повільніші. З іншого боку, користувальницький інтерфейс Home Assistant все ще виглядає дещо

недосконалим. На противагу інтерфейсу, Home Assistant в силу своєї популярності підтримує найновіші пристрої, та має відмінну підтримку медіа-пристроїв, таких як Chromecast і Roku.

Безсумнівно, обидві платформи мають численні переваги та недоліки, але вибір конкретної платформи залежить від поставленої задачі. В межах даної дипломної роботи переваги використання платформи OpenHAB переважають за переваги використання Home Assistant. OpenHAB дозволяє зручніше використовувати протокол MQTT для взаємодії з іншими пристроями, зокрема, з мікроконтроллером Arduino.

2.3. Встановлення та налаштування OpenHAB

Для завантаження OpenHAB потрібно перейти за посиланням [11] та виконати встановлення слідуючи інструкції. Після встановлення потрібно запустити сервер (рис. 2.2). Для цього потрібно перейти в каталог OpenHAB та знайти і відкрити файл **start.bat**.

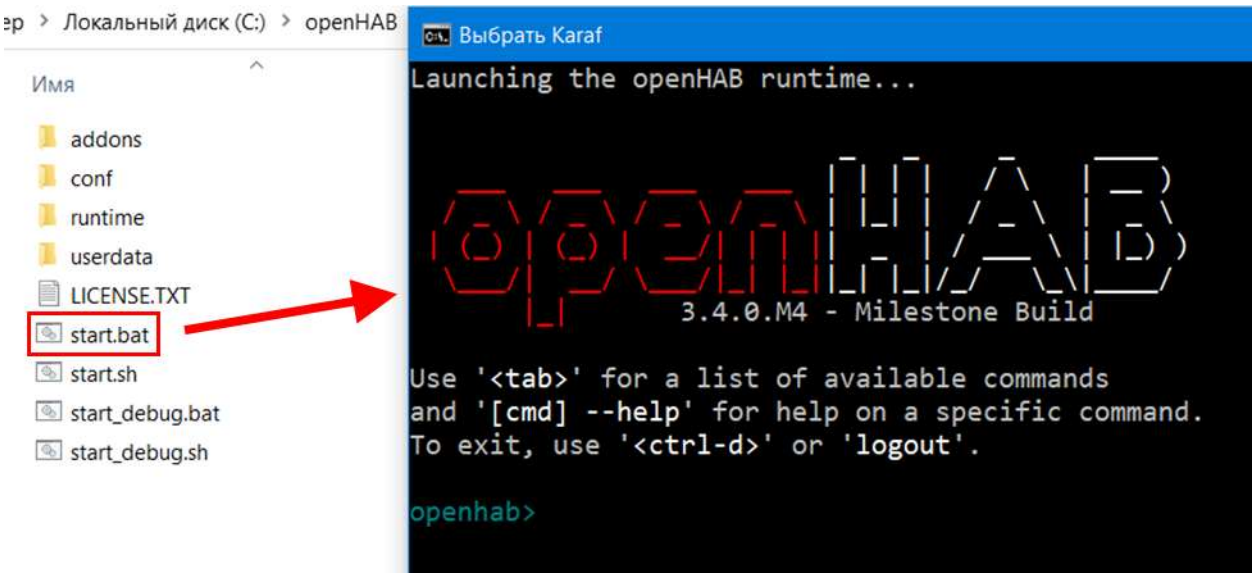

Рис. 2.2 Запуск сервера OpenHAB

Після запуску сценарію запуститься середовище виконання. Тепер потрібно запустити інтерфейс користувача, та виконати початкові налаштування [12]. Перейдемо за адресою в браузері: **http://localhost:8080**. За замовчуванням доступ до сторінок адміністрування доступний, лише якщо виконаний вхід в

обліковий запис адміністратора. Оскільки користувачів ще немає, OpenHAB просить створити обліковий запис адміністратора (рис. 2.3).

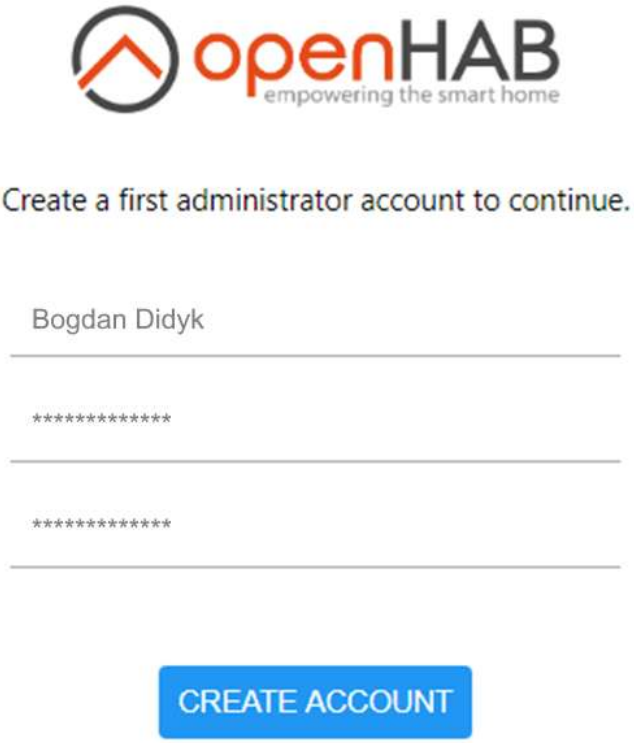

Рис. 2.3 Створення облікового запису

Після створення адміністратора, відкриється майстер початкового налаштування (рис. 2.4). Налаштування містять в собі встановлення мови, регіону та часового поясу.

Рис. 2.4 Майстер початкового налаштування

Також можна додати своє місцезнаходження (рис. 2.5) або пропустити цей крок і виконати його пізніше.

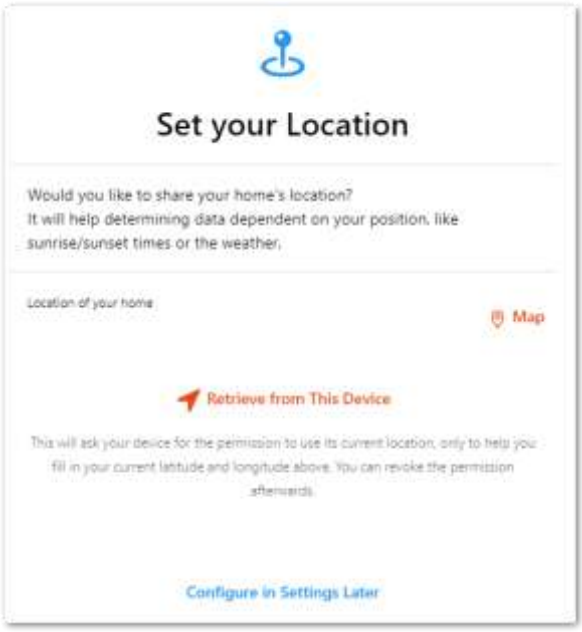

Рис. 2.5 Опис місцезнаходження

Після завершення початкових налаштувань OpenHAB перенаправить на інформаційну панель (рис. 2.6).

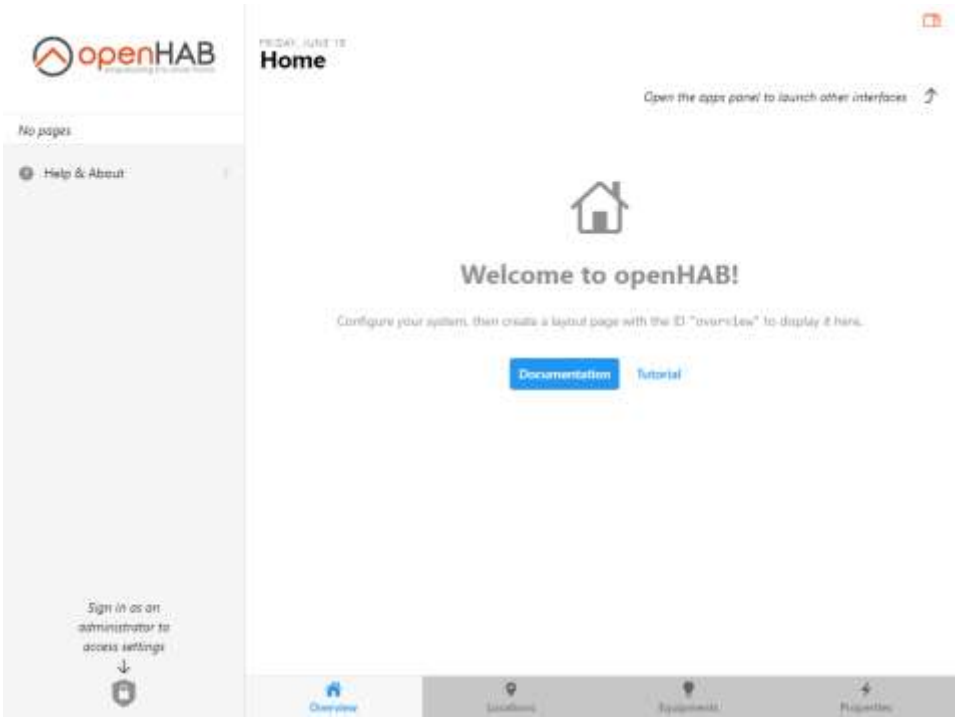

Рис. 2.6 Інформаційна панель OpenHAB

На цьому етапі вже все готово, щоб використовувати OpenHAB.

2.4. Мікроконтроллер Arduino

Arduino — це апаратна обчислювальна платформа з елементами введення / виведення для проектування, основними компонентами якої є плата мікроконтролера. На Arduino можна програмувати за допомогою мови програмування побудованої на базі C/C++. Користувач може самостійно запрограмувати реакцію компонентів системи на події, що відбуваються або використовувати вже створену бібліотеку.

Плата Arduino – це всього лише мозок пристрою (рис. 2.7). Для повноцінної роботи потрібні датчики і виконуючі пристрої. Їх підключають до плати Arduino через спеціально відведені контакти вводу-виводу.

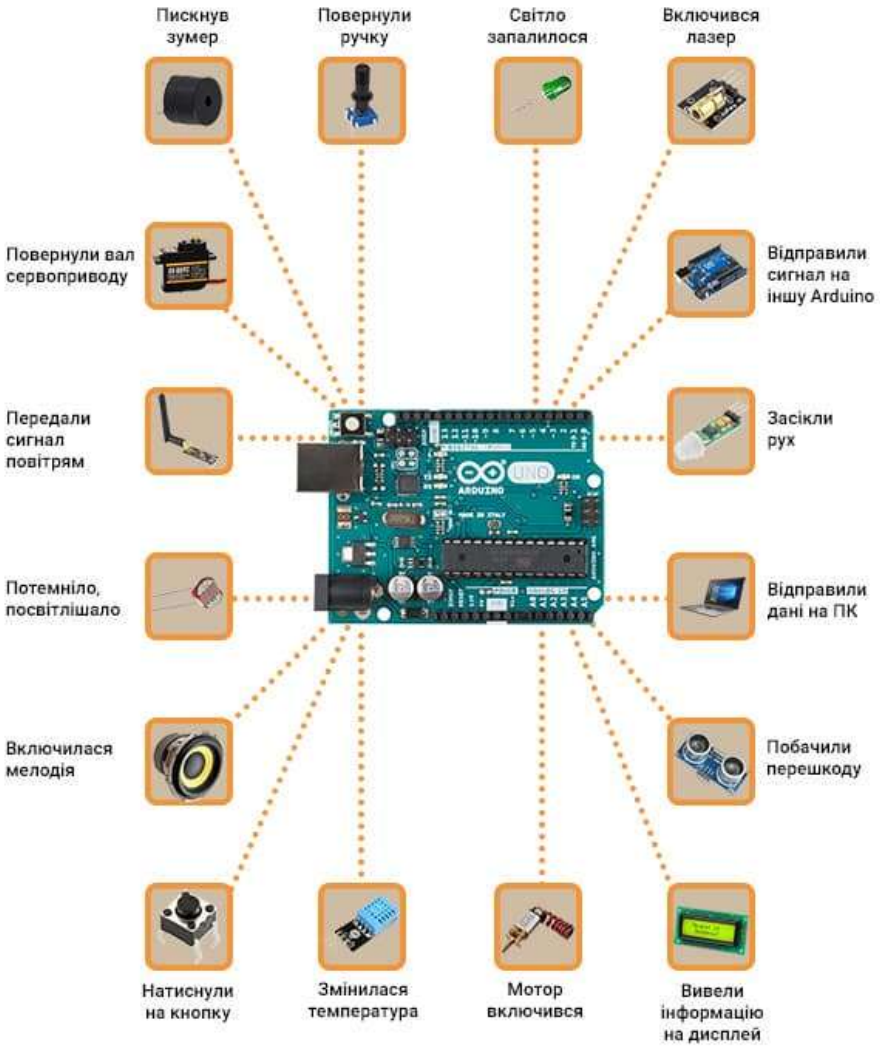

Рис. 2.7 Застосування Arduino

Платформа Arduino хороша тим, що з її допомогою можна використовувати будь-який елемент розумного будинку від різних виробників. Ця особливість дозволяє платформі не обмежуватися однією екосистемою розумного будинку, а вибирати будь-який електронний компонент для вирішення власних завдань.

Не існує розумного будинку на всі випадки життя, тому проектування починається з визначення завдань, вибору і розміщення основного вузла Arduino і інших елементів. На кінцевому етапі пов'язується і допрацьовується функціонал, за допомогою програмування. Перш ніж підбирати компоненти і модулі для створення автоматики в розумному будинку, слід приділити увагу як перевагам, так і недолікам системи [13].

Переваги IoT на базі Arduino:

- Використання компонентів інших виробників з контролером Arduino;
- Створення проекту з використанням бібліотек не вимагає багато часу;
- можна легко створення власні IoT-застосунки;
- легка маніпуляція напругою, з допомогою якої можна керувати сигналами;
- мова програмування проста, розібратися з якою зможе навіть початківець;
- можна легко переносити готові програмні рішення в Arduino, можна за допомогою USB-кабелю.

Також великою перевагою створення розумного будинку на платформі Arduino є те, що керувати системою IoT можливо за допомогою мобільних додатків через Bluetooth або Wi-Fi з’єднання.

Недоліки Arduino:

- низька частота процесора.

- порожній проект займає 466 байт на Arduino UNO і 666 байт в постійній пам'яті плати Mega;
- мала кількість флеш-пам'яті для створення програм;

Приклади IoT на Arduino [14]:

- Можна легко відстежити загублені ключі або мобільний телефон у вашому домі за допомогою Bluetooth та інших бездротових пристроїв.
- Роздрібні продавці можуть проводити A/B-тести в реальному світі за допомогою мережевих камер і датчиків, щоб визначити, як клієнти взаємодіють з конкретними продуктами та макетом магазину.
- Розумна система освітлення дозволяє місту інтелектуально забезпечувати правильний рівень освітлення, необхідний для часу доби, сезону та погодних умов. Міста продемонстрували зниження споживання енергії вуличного освітлення до 30% завдяки таким рішенням.
- Будильник може запустити кавоварку перед тим, як розбудить вас.
- Під час відпустки можна поливати свої рослини з будь-якого місця.

РОЗДІЛ 3. ПРОЕКТУВАННЯ СИГНАЛІЗАЦІЇ ТА НАЛАШТУВАННЯ УПРАВЛІННЯ НЕЮ З ДОПОМОГОЮ OPENHAB

Спроектуємо сигналізацію в межах будинку, управління якою буде здійснювати OpenHAB. Сигналізація буде працювати в real-time режимі, та її станом можна буде керувати в OpenHAB. Сама сигналізація буде створена з допомогою датчиків, які під'єднані до плати Arduino Uno. OpenHAB буде надсилати команди на плату Arduino, а сама плата Arduino в залежності від команди буде керувати датчиками.

3.1. Встановлення MQTT брокера

Використаємо перевагу OpenHAB в роботі з MQTT протоколом. Він буде протоколом зв'язку плати Ardiuno UNO з OpenHAB. Для початку потрібно встановити MQTT брокер, який буде посередником між OpenHAB та розумними пристроями. В даному випадку розумним пристроєм буде плата Arduino Uno. Встановимо Mosquitto MQTT брокер з офіційного сайту [15].

Після встановлення, потрібно подбати про безпеку. Для того, щоб підключення до брокера було захищеним, потрібно створити користувача та пароль. Для цього потрібно запустити командний рядок від імені адміністратора.

Переходимо в корінь диска C, та набираємо команду (рис. 3.1), яка створює користувача **bogdan**, після чого потрібно вказати пароль.

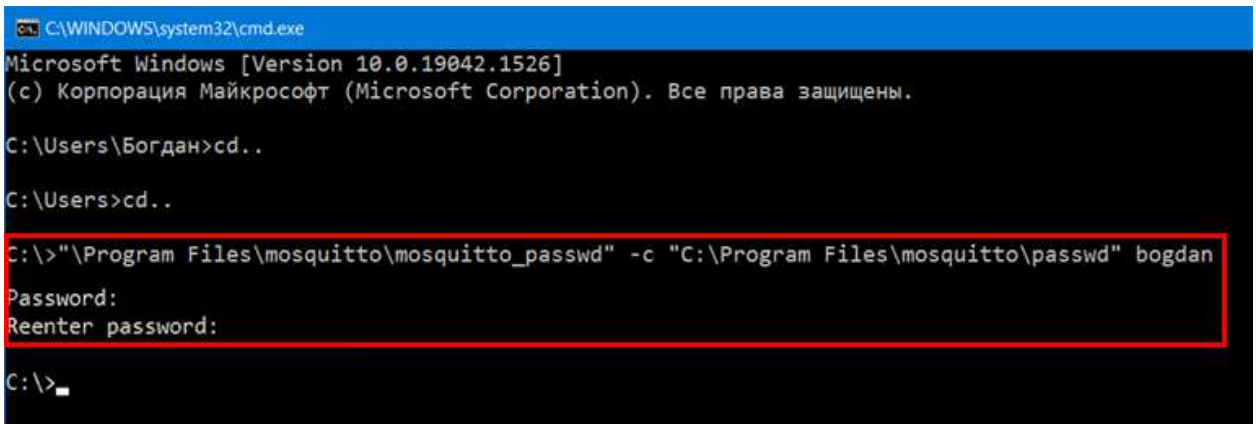

Рис. 3.1 Створення користувача та пароля

В команді вказується шлях до утиліти **mosquito_passwd** який відповідає за створення пароля.

Для того щоб переконатися, що користувач та пароль створений, потрібно перейти в каталог зі встановленим брокером, та знайти і відкрити файл **passwd** (рис. 3.2).

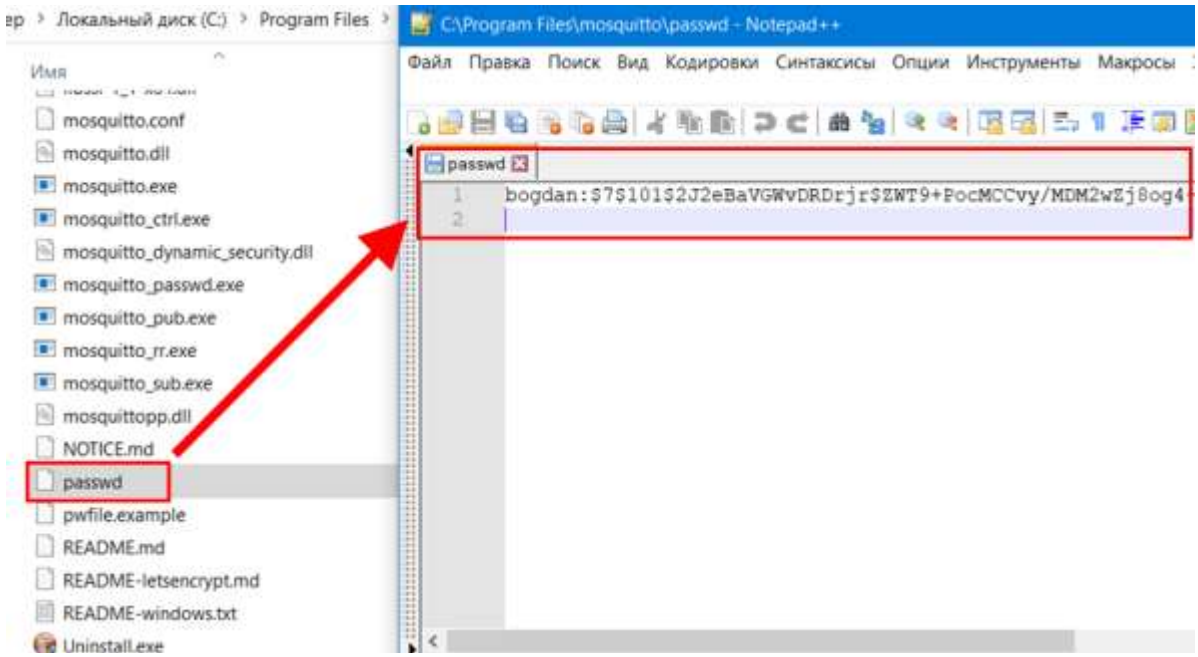

Рис. 3.2 Вміст файлу **passwd**

З рис. 3.2 видно, що файл містить назву користувача та пароль в зашифрованому вигляді.

Тепер відкриємо конфігураційний файл **mosquito.conf** (рис. 3.3) та на початку впишемо наступні команди.

```
1
2   allow_anonymous false
3   password_file %ProgramFiles%\mosquitto\passwd
4
5
6
7
8
9   # Config file for mosquitto
10  #
11  # See mosquitto.conf(5) for more information.
12  #
13  # Default values are shown, uncomment to change.
14  #
```

Рис. 3.3 Вписування команд у файл **mosquito.conf**

Перша команда вказує на заборону підключення для анонімних користувачів. Друга команда вказує брокеру шлях до файлу з паролями.

Тепер знову потрібно запустити командний рядок від імені адміністратора, та вписати команду (рис. 3.4).

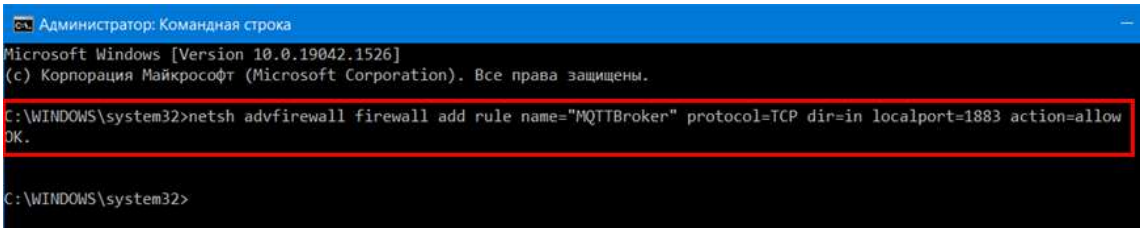

Рис. 3.4 Створення правила в брандмауері Windows

Команда з рис. 3.4 створює правило в брандмауері Windows, яке відкриє порт 1883 для вхідних підключень. Це потібно для того щоб можна було підключитися до брокера з інших пристроїв. Після виконання команди з'явився текст **OK**, що свідчить про успіх.

3.2. Налаштування зв'язку OpenHAB з MQTT брокером

Тепер налаштуємо сполучення OpenHAB з MQTT брокером. Налаштування повністю відбуваються в OpenHAB.

Для початку потрібно встановити модуль MQTT Binding. Для цього переходимо в меню **Settings** та в розділі **Add-ons** вибираємо меню **Bindings** (рис. 3.5).

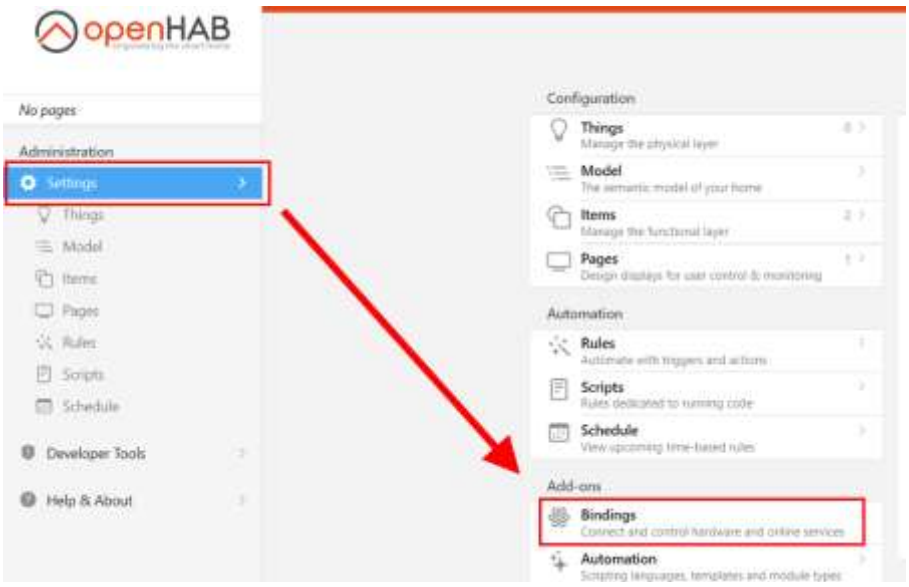

Рис. 3.5 Меню **Settings** в OpenHAB

В меню шукаємо необхідний модуль (рис. 3.6).

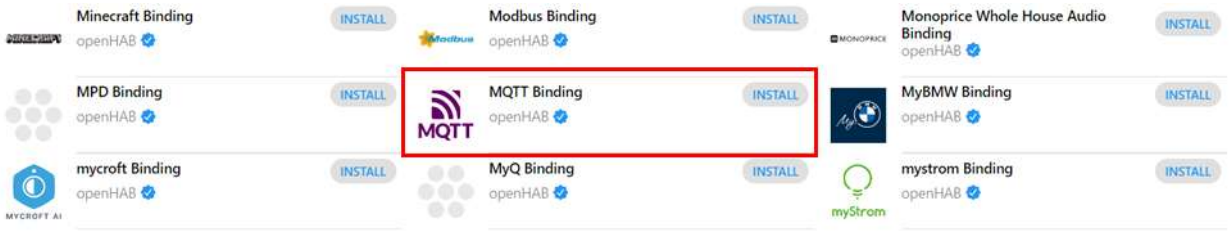

Рис. 3.6 Список модулів OpenHAB

Встановлюємо його (рис 3.7), клацнувши на нього а потім на кнопку **Install**.

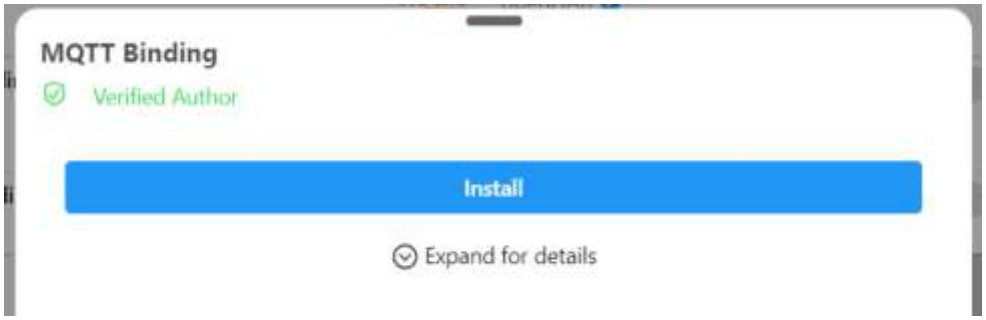

Рис. 3.7 Встановлення модуля MQTT Binding

Модуль вже встановлений, але ще потріно створити та налаштувати компонент MQTT Broker. Для цього переходимо в меню **Things** та обираємо щойно встановлений модуль MQTT Binding (рис. 3.8).

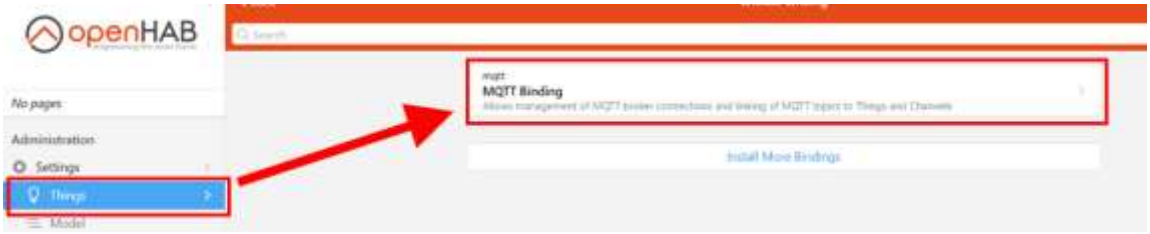

Рис. 3.8 Модуль MQTT Binding в меню **Things**

Далі потрібно вибрати MQTT Broker (рис. 3.9).

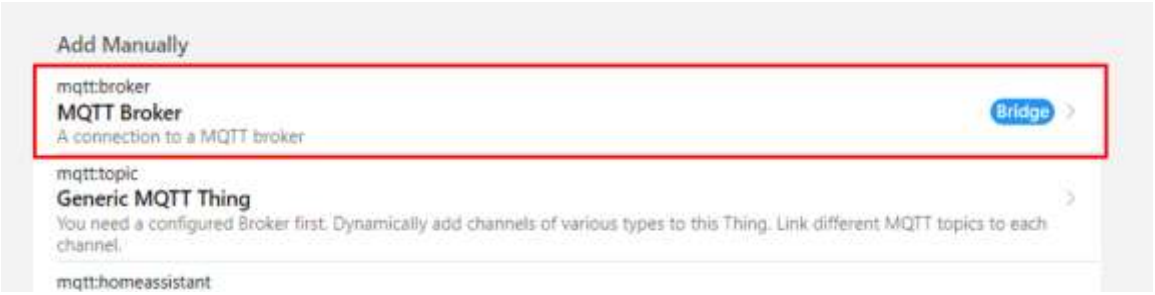

Рис. 3.9 Вибір модуля MQTT Binding для зв'язку з брокером

Після цього відкриються налаштування (рис 3.10). Спочатку потрібно вписати унікальний ідентифікатор та IP-адресу брокера.

Рис. 3.10 Налаштування MQTT брокера

Натискаємо на **Show advanced** щоб відкрити додаткові налаштування. Шукаємо область де треба вписати логін та пароль (рис. 3.11). Логін та пароль має бути той, який був вказаний при встановленні брокера.

Рис. 3.11 Вказування користувача та пароля брокера

Після цього натискаємо на кнопку **Create Thing**. Підключення тепер активне зі статусом **ONLINE** (рис 3.12).

Рис. 3.12 Налаштований MQTT брокер зі статусом ONLINE

Тепер OpenHAB має сполучення з MQTT брокером.

3.3. Налаштування розумного пристрою в OpenHAB

Добавимо новий пристрій. Для цього в меню **Things** створюємо новий пристрій та обираємо модуль **MQTT Binding**. Далі обираємо **Generic MQTT Thing** (рис. 3.13).

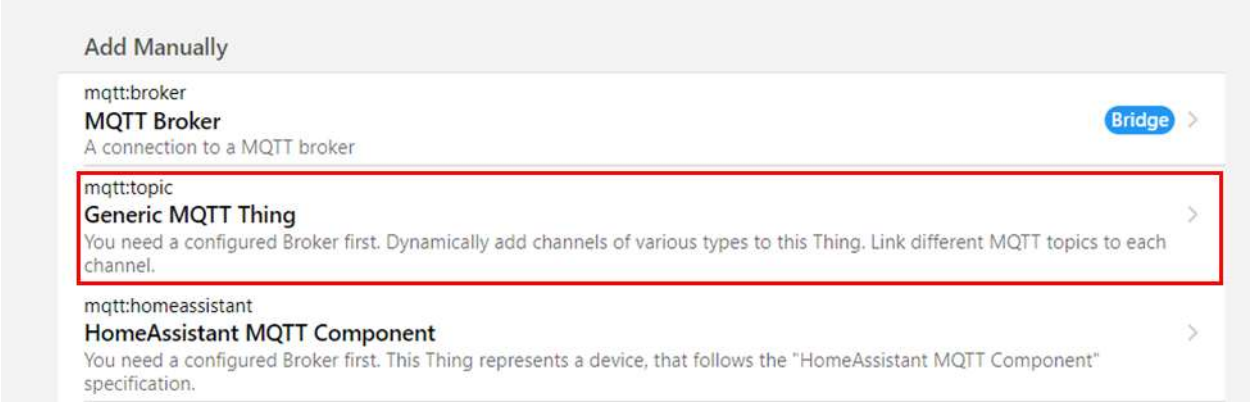

Рис. 3.13 Додавання нового пристрою

Вписуємо ID, опис, та вказуємо міст **MQTT Broker**. Після цього тиснемо на кнопку **Create Thing** (рис 3.14).

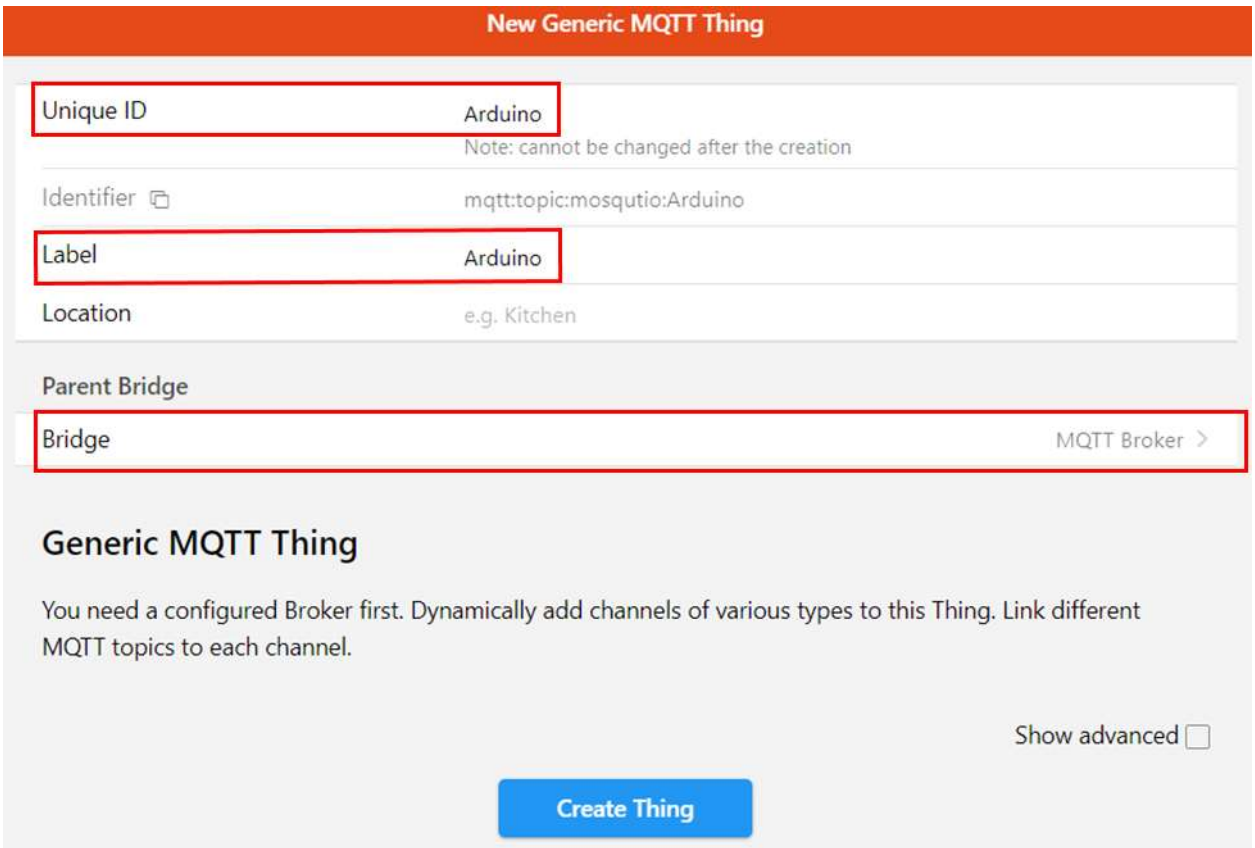

Рис. 3.14 Налаштування пристрою

Після цього в розділі **Things** з'явився щойно створений пристрій (рис. 3. 15).

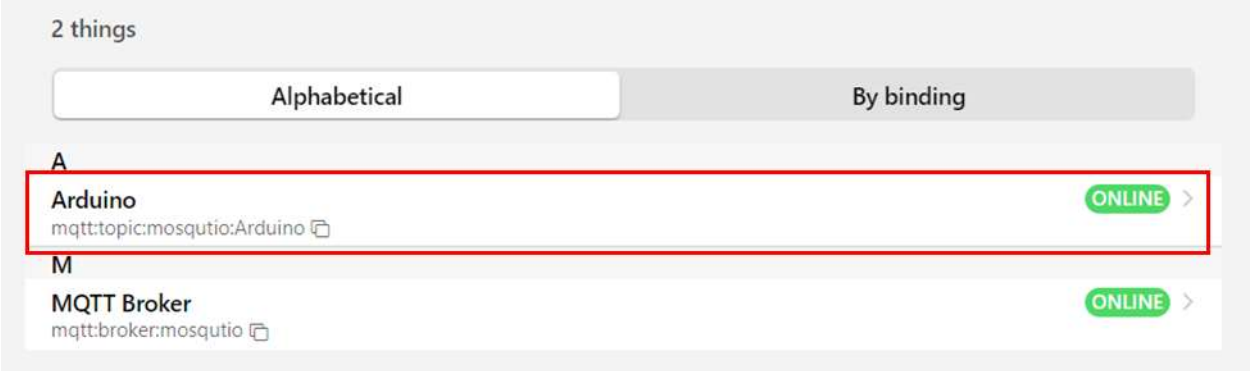

Рис. 3.15 Створений пристрій зі статусом ONLINE

Пристрій налаштований зі статусом **ONLINE.**

В даному випадку розумний пристрій — плата Arduino. В OpenHAB потрібно налаштувати канал для взаємодії зі станом сигналізації.

3.4. Створення каналу взаємодії зі станом сигналізації

Налаштуємо канал для Arduino (рис 3.16). Клікаємо на елемент та відкриваємо вкладку **Channels**. Натискаємо меню **Add Channel**. Вписуємо назву, опис та обираємо тип каналу «перемикач».

Рис. 3.16 Налаштування каналу

Також потрібно налаштовувати команду яка OpenHAB буде відправляти Arduino для керування перемикачем (рис. 3.17). Встановлюємо значення 1 для стану "включено" та 0 в протилежному випадку.

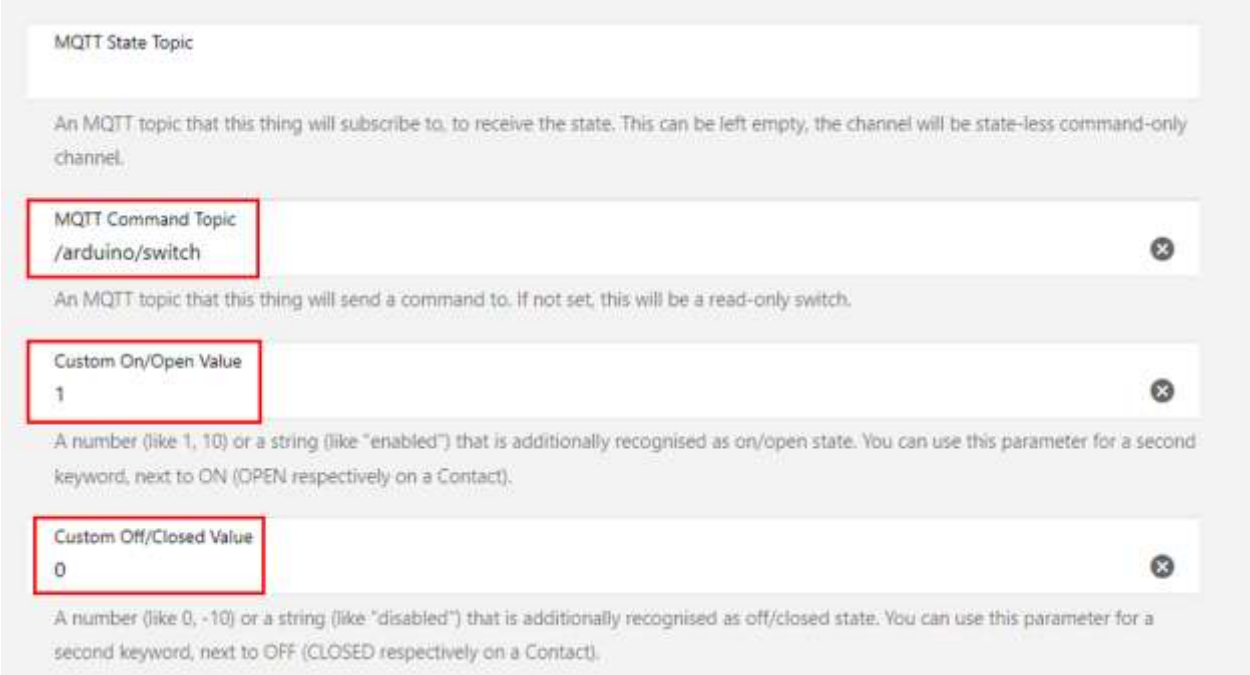

Рис. 3.17 Налаштування команди для керування перемикачем

На цьому етапі вже можна керувати станом вимикача, але для зручності краще спочатку налаштувати інтерфейс, з допомогою якого буде виконуватися керування.

3.5. Налаштування інтерфейсу для керування станом

На основі створеного каналу потрібно створити елемент інтерфейсу (рис. 3.18). Для цього натискаємо на пункт **Add Equipment to Model**.

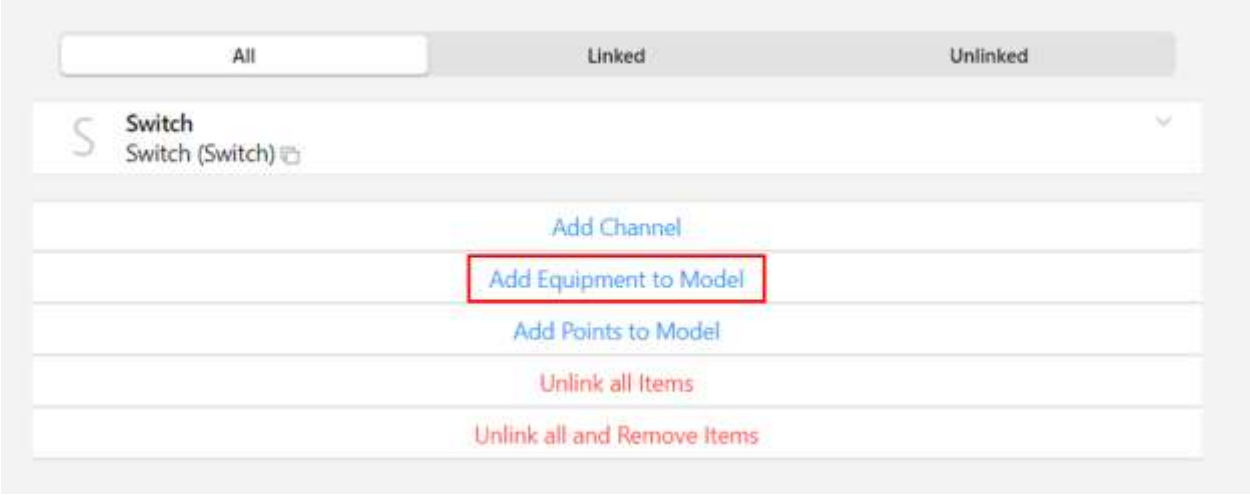

Рис. 3.18 Створення елементу інтерфейсу

Далі створюємо ім'я, опис та категорію (рис. 3.19).

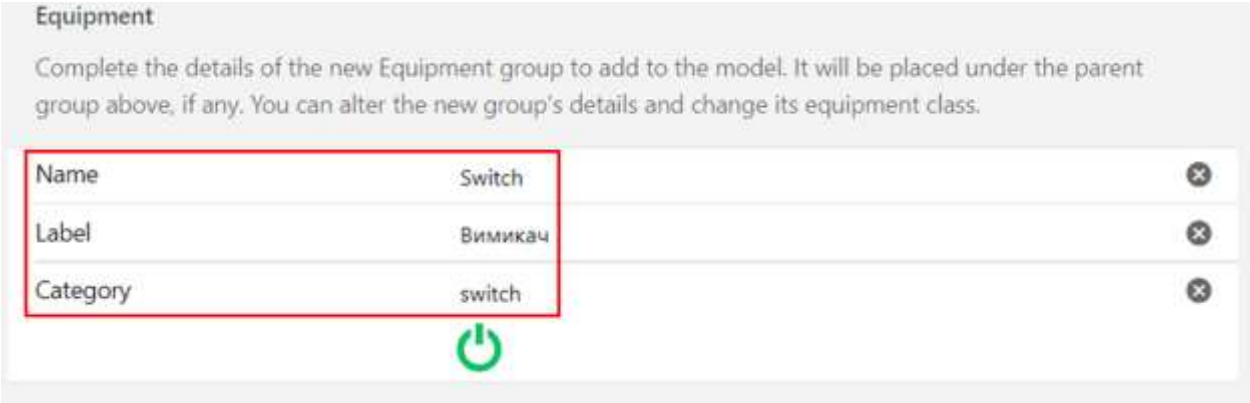

Рис. 3.19 Створення імені, опису та категорії

Прогортаємо далі, та вибираємо канал, та основі якого буде створений елемент інтерфейсу (рис. 3.20). Також встановлюємо назву, опис та категорію. Після цього вибираємо семантичний клас **Switch**, та семантичну властивість **Light**.

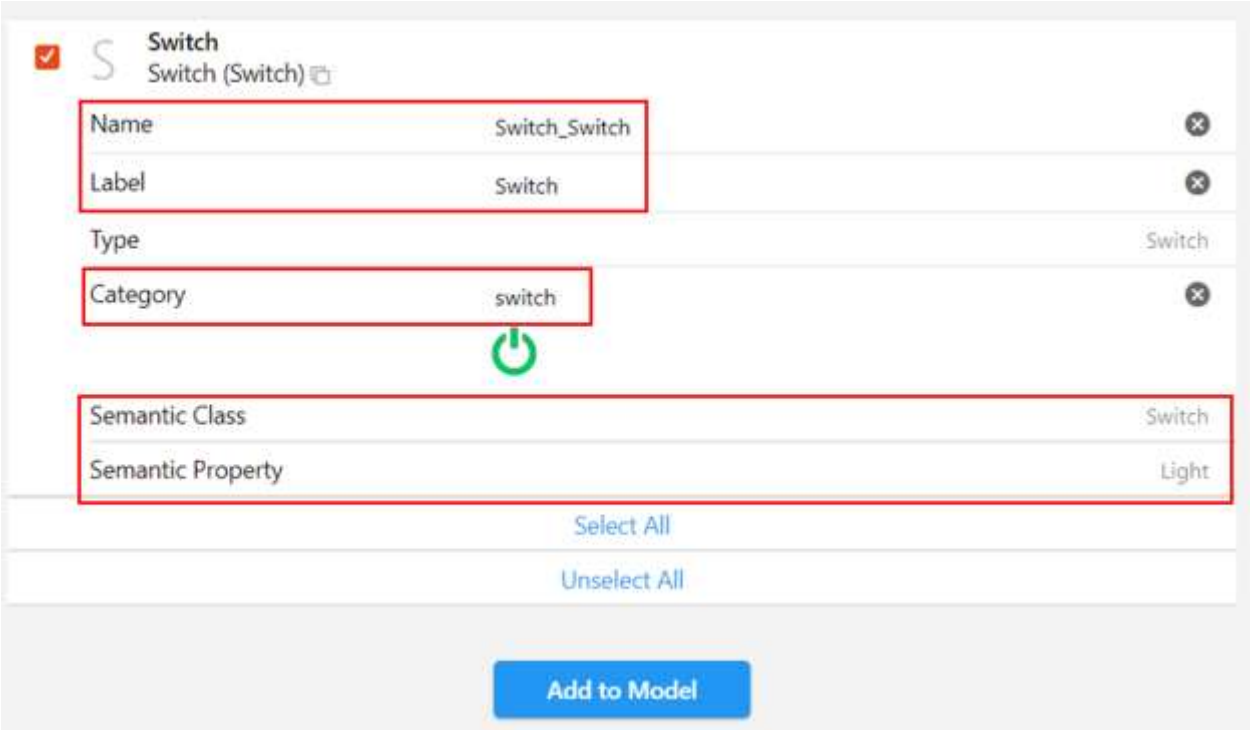

Рис. 3.20 Вибір каналу **Switch**, на основі якого буде створено елемент інтерфейсу

Після цих налаштувань елемент інтерфейсу автоматично з'єднаний з каналом.

Тепер потрібно перейти в розділ **BasicUI** (рис. 3.21).

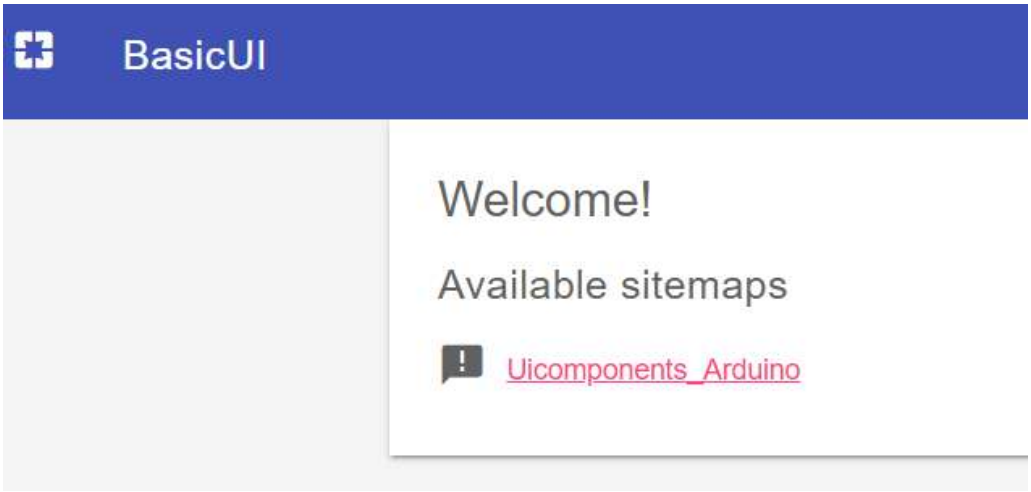

Рис. 3.21 Розділ BasicUI

Тут доступний розділ **Uicomponents_Arduino** (рис. 3.22).

Рис. 3.22 Елементи управління розділу **Uicomponents_Arduino**

Як видно з рисунку, візуальний компонент створений.

Протестуємо інтерфейс з мобільного пристрою. Для цього відкриваємо браузер, та вводимо в адресний рядок IPv4-адресу та вказуємо порт 8080. Після цього в OpenHAB переходимо в розділ **BasicUI** (рис. 3.23).

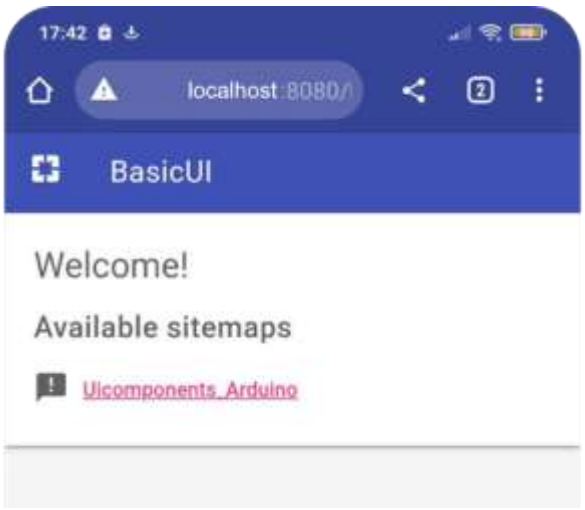

Рис. 3.23 Розділ BasicUI на смартфоні

Далі відкриваємо **Uicpomponents_Arduino** (рис. 3.24).

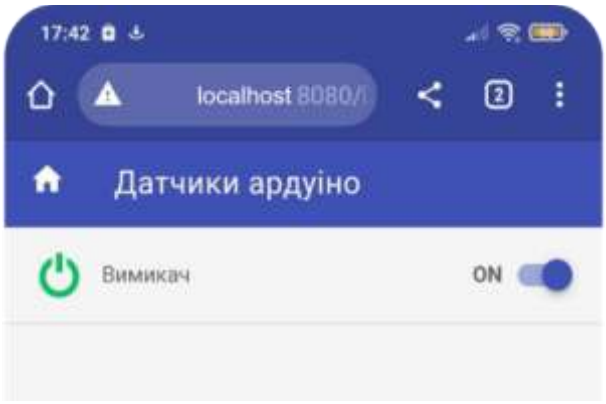

Рис. 3.24 Елементи управління розділу **Uicomponents_Arduino** на смартфоні

Як і в ПК версії OpenHAB, для нас доступний інтерфейс, з допомогою якого можна керувати створеним каналом.

3.6. Проектування схеми та написання коду в Arduino IDE

Схема сигналізації з використанням датчиків Arduino показана на рисунку 3.25.

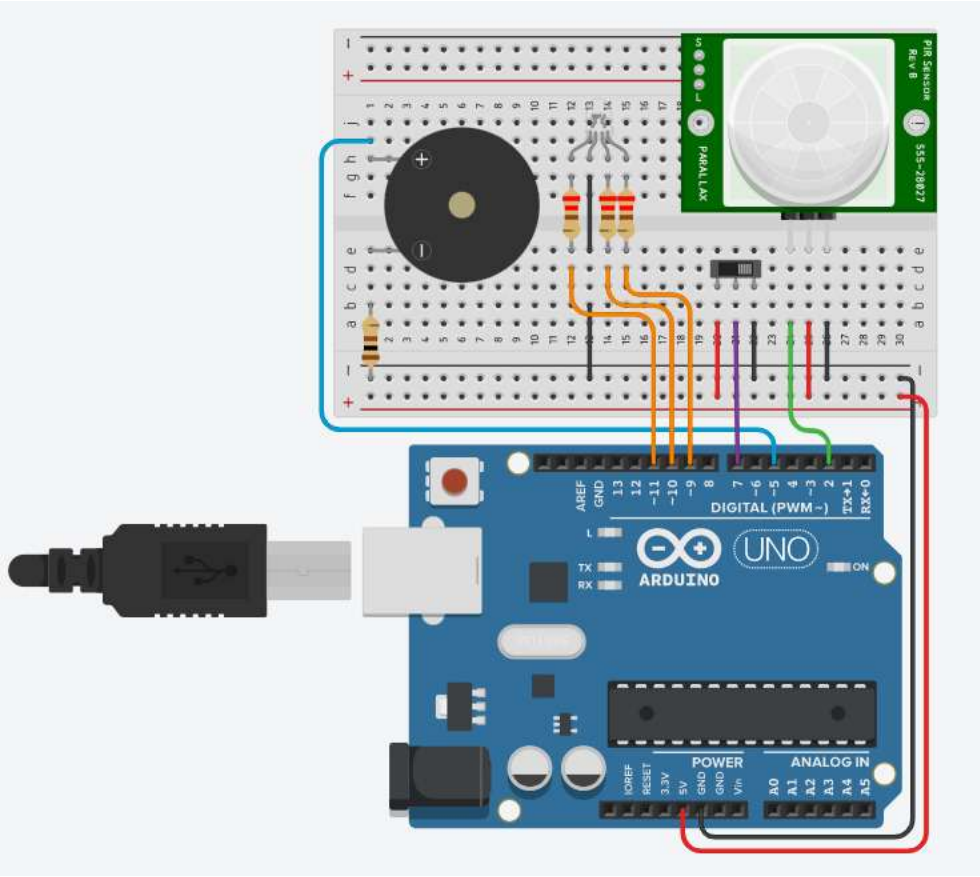

Рис. 3.25 Схема сигналізації на Arduino

Схема містить такі компоненти:

- Arduino Uno;
- макетна плата;
- п'єзоелемент для відтворення звукового сигналу;
- RGB світлодіод;
- датчик руху PIR1;
- перемикач;
- резистор на 100 Ом для п'єзоелемента;
- три резистора на 220 Ом для входів RGB світлодіода.

П'єзоелемент та RGB світлодіод слугують для візуально-звукового відтворення при спрацюванні сигналізації: світлодіод мигає червоним кольором, а п'єзоелемент відтворює звуковий сигнал. датчик руху PIR1 потрібен для виявлення руху в області його видимості. Якщо датчик фіксує рух, то він повертає сигнал в 5В. Для програмування це зручно, оскільки не потрібно виконувати додаткові операції з вхідним сигналом (як у випадку з аналоговим). Посилання на онлайн-проект для демонстрації роботи сигналізації https://www.tinkercad.com/things/atDw5j8lNQ9.

Тепер потрібно написати програму (скетч) та завантажити її на плату Arduino. Для цього є середовище Arduino IDE. Потрібно перейти на офіційний сайт [16] встановити на ОС Windows. Після встановлення, відкриваємо середовище. Спочатку для нього середовищу додаткову інформацію, а саме, модель плати Arduino (рис. 3.26). В даному випадку це Arduino Uno.

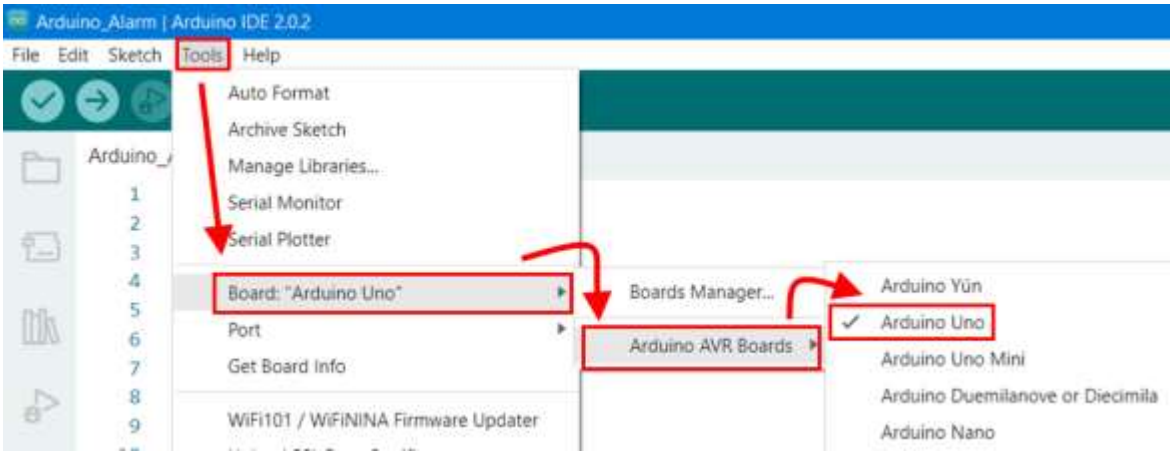

Рис. 3.26 Вибір моделі Arduino в середовищі Arduino IDE

Тепер вписуємо код (Додаток А) в середовище (рис. 3.27).

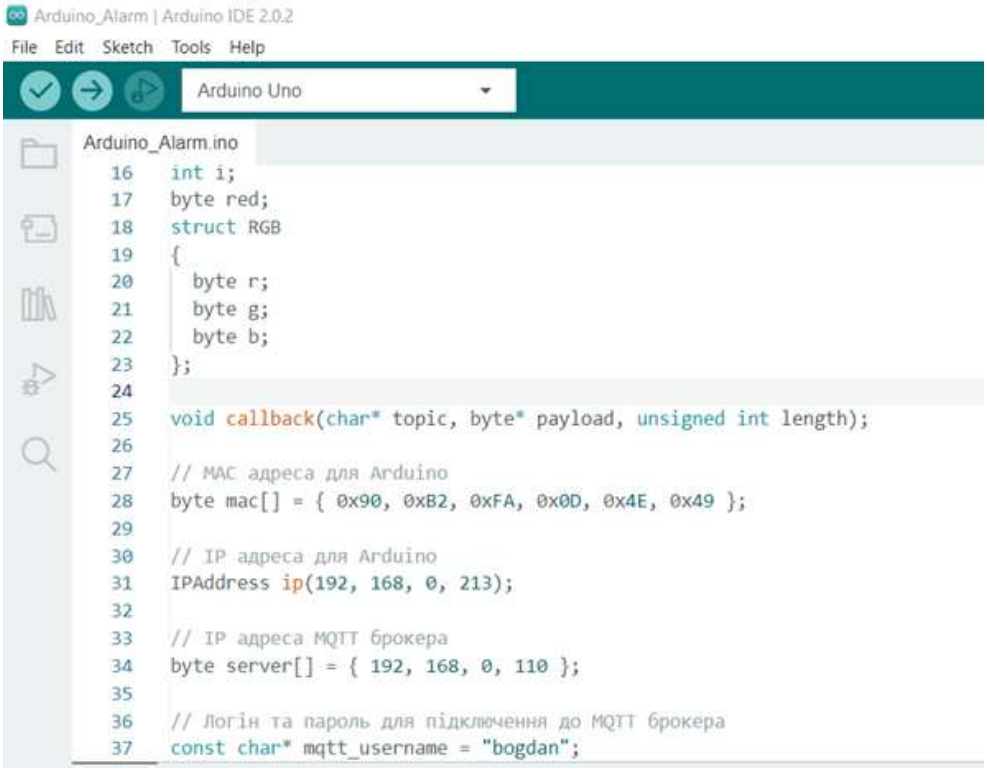

Рис. 3.27 Код в середовищі Arduino IDE

В коді, окрім алгоритму роботи сигналізації, необхідно вказати MAC-адресу та IPv4-адресу для Arduino та IPv4-адресу на якій розміщений MQTT брокер.

На цьому етапі потрібно під'єднати Arduino Uno через USB-кабель, та вибрати той COM-порт, який з'явився разом з підключенням (рис. 3.28). В даному випадку це порт COM3.

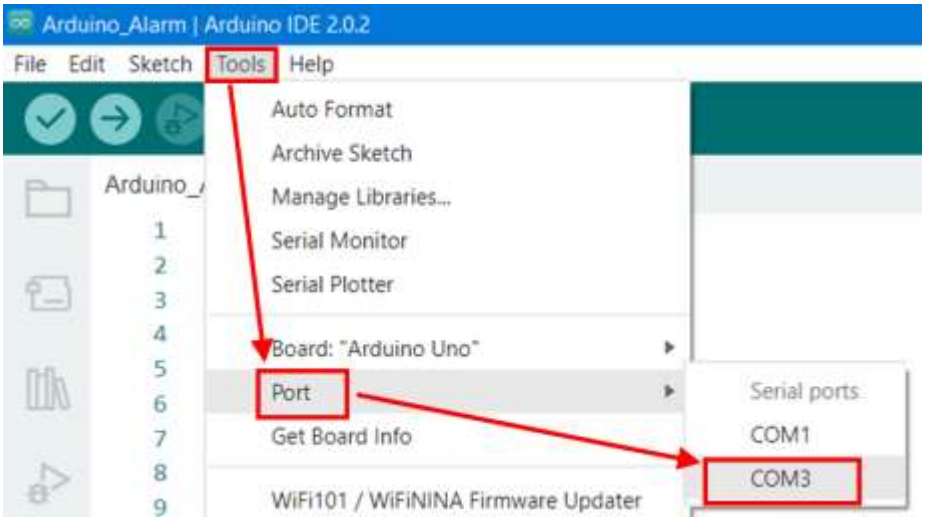

Рис. 3.28 Вибір COM-порта, який з'язвився з підключенням Arduino

Тиснемо кнопку Upload, після чого скетч буде автоматично скомпільовано та завантажено на плату Arduino.

Тепер можна протестувати сигналізацію в ділі. В OpenHAB на вкладці BasicUI перемикаємо кнопку з вимкненого стану в увімкнений. Після цього сигналізація розпочала свою роботу. Якщо в область видимості датчика попадеться якийсь рух, то спрацює сигналізація. В Arduino IDE відкриємо монітор порта (рис. 3.29).

Рис. 3.29 Вивід інформації в монітор порта

Як видно з рисунку 3.29, текст містить інформацію стосовно роботи скетчу. Спочатку йде під'єднання до створеного каналу OpenHAB через MQTT брокер, а потім виведення стану натиснутої кнопки в BasicUI.

ВИСНОВКИ

В ході виконання магістерської роботи були виконані наступні завдання дослідження:

- ☑ Проведено огляд Інтернету речей, її екосистеми та архітектури:
- ☑ розглянуто програмний комплекс OpenHAB з його перевагами та мікроконтроллер Arduino;
- ☑ реалізовано керування платою Arduino з використанням OpenHAB.

Без сумніву можна сказати, що Інтернет речей — це перспективний розвиток Інтернету на найближчі десятиліття. Бурхливий розвиток IoT-пристроїв позитивно вплине на стан інформаційних технологій, та зокрема на всі сфери життя людини.

В ході виконання виконання поставлених завдань, я зрозумів, що застосування OpenHAB для реалізації системи керування IoT-пристроїв є чудовим вибором. З його допомогою можна легко спроектувати систему домашньої автоматизації без великих технічних знань. Однак, OpenHAB можна застосовувати для проектування систем автоматизації не лише в межах будинку, а й в межах району, підприємства, міста тощо в силу його гнучкої масштабованості. Зокрема, використання всіх переваг OpenHAB дозволяє досягти високої ефективності та гнучкості при проектуванні таких систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поняття Інтернету речей. https://bit.ly/3EStJ8H
2. Прогнози щодо кількості підключених IoT-пристроїв до мережі Інтернет. https://bit.ly/3v7vGIC
3. Застосування Інтернету речей. https://bit.ly/3VhoIvL
4. Екосистема Інтернету речей. https://bit.ly/3TZGJO2
5. Архітектура Інтернету речей. https://bit.ly/3V0ddcz
6. Розумний будинок та його функції. https://bit.ly/3Eu2wbd
7. Вимоги до розумного будинку. https://bit.ly/3ABcu9s
8. Загальний опис OpenHAB. https://bit.ly/2U3POar
9. Порівняння OpenHAB та Home Assistant. https://bit.ly/3jmPVzB
10. Офіційний сайт проекту OpenHAB. https://bit.ly/3FRjVMC
11. Інструкція по встановленні OpenHAB на ОС Windows. https://bit.ly/3j4u6V8
12. Налаштування середовища OpenHAB. https://bit.ly/3BTMLtD
13. Інтернет речей на Arduino. https://bit.ly/3hAgcts
14. Приклади Інтернету речей на Arduino. https://bit.ly/3E09ioD
15. Офіційний сайт проекту Mosquitto. https://bit.ly/2HyTDRY
16. Офіційний сайт середовища Arduino IDE. https://bit.ly/3WDQovo

ДОДАТКИ

Додаток А. Скетч для Arduino

```
#include <PubSubClient.h>
#include <SPI.h>
#include <Ethernet.h>

#define RED_PIN 11
#define GREEN_PIN 9
#define BLUE_PIN 10
#define SLIDER_PIN 7
#define SENSOR_PIN 2
#define BUZZER_PIN 5
#define MIN_FREQUENCY 1000
#define MAX_FREQUENCY 1400
#define COUNT 3

boolean state = HIGH;
int i;
byte red;
struct RGB
{
  byte r;
  byte g;
  byte b;
};

void callback(char* topic, byte* payload, unsigned int length);

// MAC адреса для Arduino
byte mac[] = { 0x90, 0xB2, 0xFA, 0x0D, 0x4E, 0x49 };

// IP адреса для Arduino
IPAddress ip(192, 168, 0, 213);

// IP адреса MQTT брокера
byte server[] = { [IPv4 адреса] };

// Логін та пароль для підключення до MQTT брокера
const char* mqtt_username = "[Користувач]";
```

```
const char* mqtt_password = "[Пароль]";

EthernetClient ethClient;
PubSubClient client(server, 1883, callback, ethClient);

// Підключення та підписка на MQTT брокер
boolean reconnect()
{
  Serial.println("reconnect...");
  if (client.connect("Switch", mqtt_username, mqtt_password))
  {
    client.subscribe("/arduino/switch");
    Serial.println("Connected to: /arduino/switch");
    Serial.println("MQTT connected");
  }
  return client.connected();
}

void setup()
{
  Serial.begin(9600);
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
  pinMode(SLIDER_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  Ethernet.begin(mac, ip);

  Serial.print("My ip address: ");
  Serial.println(ip);

  reconnect();
}

void loop()
{
  client.loop();
```

```
  if(digitalRead(SLIDER_PIN) && digitalRead(SENSOR_PIN))
  {
    Serial.println("Alarm activated!");
    activateAlarm();
  }
  else
  {
    setRGBColor({0, 255, 0});
  }
}

// Читання даних з MQTT брокера
void callback(char* topic, byte* payload, unsigned int length)
{
  Serial.println("callback...");
  // Перевірка нових повідомлень в підписках брокера
  payload[length] = '\0';
  Serial.print("Topic: ");
  Serial.print(String(topic));
  Serial.println(" – ");

  if (String(topic) == "/arduino/switch")
  {
    String value = String((char*)payload);
    state = value.substring(0, value.indexOf(';')).toInt();
    Serial.print("Value: ");
    Serial.println(state);
    digitalWrite(SLIDER_PIN, state);
  }
}

// Функція, яка активовує сигналізацію
void activateAlarm()
{
  for(int k = 0; k <= COUNT; k++)
  {
    for(i = MIN_FREQUENCY; i < MAX_FREQUENCY; i++)
    {
      activateBuzzerAndLED(i);
```

```
    }
    for(i = MAX_FREQUENCY; i > MIN_FREQUENCY; i--)
    {
      activateBuzzerAndLED(i);
    }
  }
}

// Функція, яка створює візуально-звукову складову
void activateBuzzerAndLED(int signal)
{
  red = map(signal, MIN_FREQUENCY, MAX_FREQUENCY, 0, 255);
  setRGBColor({255, red, red});
  tone(BUZZER_PIN, signal, 5);
  delay(2);
}

// Функція, яка налаштовує канал RGB для світлодіода
void setRGBColor(struct RGB color)
{
  analogWrite(RED_PIN, color.r);
  analogWrite(GREEN_PIN, color.g);
  analogWrite(BLUE_PIN, color.b);
}
```