

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ

завідувач кафедри

мережевих та інтернет технологій

_____ **Юрій КРАВЧЕНКО**

«_____» _____ 2022 року

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

галузі знань 17 «Електроніка та телекомунікації»
за спеціальністю 172 «Телекомунікації та радіотехніка»
освітньо-професійна програма «Мережеві та інтернет технологій»

на тему:

**Інформаційна система сповіщення тривоги чи
надзвичайної ситуації засобами ІОТ**

Виконав: студент групи МІТ -41

_____ **Микола Прохоренко**

(ім'я та ПРІЗВИЩЕ)

_____ (підпис)

Керівник: доцент кафедри мережевих та інтернет технологій

(посада)

_____ **к.т.н., доцент Ольга ЛЕЩЕНКО**

(науковий ступень, вчене звання, ім'я та ПРІЗВИЩЕ)

_____ (підпис)

Київ 2022

Міністерство освіти і науки України
«Київський Національний університет імені Тараса Шевченка»
Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ
 завідувач кафедри
 мережевих та інтернет технологій
 _____ Ю.В. Кравченко
 « ____ » _____ 2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ

Здобувачу вищої освіти _____ Прохоренку Миколі Володимировичу
 (прізвище, ім'я, по батькові)

1. Тема роботи:
 Інформаційна система сповіщення тривоги чи надзвичайної ситуації
 засобами ІОТ

затверджена на засіданні кафедри МІТ «24» грудня 2021 р. протокол №8

2. Термін здачі закінченої роботи «30» травня 2022р

3. Вихідні дані до проекту
 (роботи)

Мова програмування – С, С#.

Технологія Arduino: контролер, датчики, bluetooth-модуль. Програмне забезпечення XCSoar.

4. Зміст пояснювальної записки (перелік питань, що їх потрібно розробити, обсяг – 35-40 стор.)

Вступ

1. Аналіз технологій інтернету речей.

1.1. Історія технології «Інтернет речей». Що таке ІоТ

1.2. Компоненти архітектури ІоТ.

1.2.1 Архітектура ІоТ

1.2.2 Програмне забезпечення ІоТ

1.3. Передача даних Інтернет речей (ІоТ)

2. Апаратне забезпечення інформаційної системи

2.1. Компоненти ІоТ

2.2. Загальний опис системи

2.3. Налаштування мережевих параметрів

3. Розробка програмного забезпечення інформаційної системи

3.1 Апаратна частина інформаційної системи

3.2. Інструменти розробки

3.3 Дослідження інформаційної системи

3.4 Тестування системи

Висновки

5. Перелік графічного матеріалу 8-10 слайдів

Дата видачі завдання

Керівник роботи

Ольга ЛЕЩЕНКО

(підпис)

(посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання

Микола ПРОХОРЕНКО

(підпис)

Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Підготовчий	02.05.2022	
2	Розділ 1	10.05.2022	
3	Розділ 2	15.05.2022	
4	Розділ 3	20.05.2022	
5	Доповідь та слайди	25.05.2022	
6	Пояснювальна записка	30.05.2022	

Здобувач вищої освіти _____ Микола ПРОХОРЕНКО
(підпис)

Керівник _____ Ольга ЛЕЩЕНКО
(підпис)

РЕФЕРАТ

Пояснювальна записка: 46 с., 35 рис., 1 табл. , 8 джерел.

Об'єкт дослідження: система сповіщення небезпеки.

Предмет дослідження: інформаційна система сповіщення тривоги чи надзвичайної ситуації засобами ІОТ.

Методи дослідження: симуліція системи в програинному середовищі Cisco Packet Tracer.

Мета даної дипломної роботи полягає в розв'язання задачі сповіщення тривоги чи надзвичайної ситуації.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- Налаштування девайсів та з'єднання з домашнім шлюзом;
- Програмування плат Arduino для правильної роботи датчиків;
- Розробка фізичної моделі системи ІоТ.

Розроблено фізичну модель інформаційної системи. Виконано проектування інформаційної системи сповіщення надзвичайної ситуації та її побудову.

Здійснено програмну реалізацію налаштування датчиків та забезпечено керування девайсами через програму ІоТ Monitor.

Практичне значення одержаних результатів полягає у забезпеченні інформування населення про потенційні небезпеки. Матеріали розробки також можуть слугувати основою для подальшого удосконалення системи сповіщення тривоги чи надзвичайної ситуації.

Наукова новизна дослідження полягає у створенні інформаційної системи на базі плат Arduino для сповіщення надзвичайної ситуації.

Галузь використання – приватні будинки та квартири.

Напрямки подальших досліджень: удосконалена система автоматичного сповіщення з вказанням загрози.

Ключові слова:

ІОТ, РОЗУМНИЙ ДІМ, ДАТЧИКИ, СЕНСОРИ, ШЛЮЗ, ARDUINO, CISCO PACKET TRACER, ІОТ MONITOR.

ЗМІСТ

ВСТУП	7
1 Аналіз технологій «Інтернет речей»	9
1.1 Історія технології «Інтернет речей»	9
1.2 Компоненти архітектури IoT.....	11
1.2.1 Архітектура IoT.....	13
1.2.2 Програмне забезпечення IoT.....	15
1.3. Передача даних Інтернет речей (IoT)	17
2. Апаратне забезпечення інформаційної системи.....	20
2.1.Компоненти IoT.....	20
2.2.Загальний вигляд системи.....	27
2.3. Налаштування мережевих параметрів.....	29
3. Розробка програмного забезпечення інформаційної системи.....	34
3.1 Апаратна частина інформаційної системи.....	34
3.2. Інструменти розробки.....	35
3.3 Дослідження інформаційної системи.....	41
3.4 Тестування системи.....	45
Висновок.....	50
Перелік посилань.....	52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IoT - Internet of Things

LAN - Local Area Network

MCU - Multipoint Control Unit

ІС - інформаційна система

TCP - протокол керування передаванням

IP - міжмережевий протокол

ТЗ - технічне завдання

ВСТУП

У всьому світі спостерігається широке впровадження технологій Інтернет речей (IoT). Розумна побутова техніка, переносні прилади, датчики та інші смартпристрої стають масовими. Популярність розумних споживчих пристроїв очікується й на далі та буде зростати шаленими темпами.

Сучасні інформаційні технології пронизують всі сфери суспільного життя. Під їх впливом відбувається повна зміна всіх процесів в системі суспільного управління, вони сприяють ефективному виробленню управлінських рішень та автоматизують багато процесів.

Муніципалітети у всьому світі приймають на озброєння IoT, працюючи над тим, щоб стати розумними містами, які покладаються на дані, отримані різноманітними датчиками, розповсюджених по певному географічному регіону. За допомогою яких всі мешканці міста які будуть підключені до даної мережі матимуть доступ до всіх потрібних даних, що своєю чергою сприятиме полегшенню роботи та побуту людей.

IoT є наступним кроком на шляху до діджиталізації сучасного суспільства, де предмети й люди пов'язані один з одним через комунікаційні мережі та з'являється можливість повідомляти про їх стан та стан навколишнього середовища.

Дивлячись за останніми подіями які відбуваються в Україні все більш актуальною становиться використання IOT для забезпечення власної безпеки та сповіщення населення про потенційні загрози. Що б могло спростити інформування населення та зберегло багато людських життів.

Наприклад в Ізраїлі є служба тилу, яка має сайт і додаток з усіма інструкціями — що потрібно робити під час обстрілів, землетрусів, де знаходиться найближче бомбосховище і т.д. В разі конфлікту за допомогою різноманітних пристроїв служба отримує правдиві дані та доносить інформацію будь-якими доступними способами населенню: через смс-повідомлення, телебачення, інтернет, радіо.

На цей час в Україні не існує такої служби та побудованої мережі інформування, але беручи до уваги досвід Ізраїлю ми маємо можливість впровадити дану систему в кожен будинок та стати на один крок ближче до повної цифровізації.

1 АНАЛІЗ ТЕХНОЛОГІЇ «ІНТЕРНЕТ РЕЧЕЙ»

1.1 Історія технології «Інтернет речей»

Інтернет речей (IoT) описує фізичні об'єкти (або групи таких об'єктів) із датчиками, можливостями обробки, програмним забезпеченням та іншими технологіями, які підключаються та обмінюються даними з іншими пристроями та системами через Інтернет чи інші комунікаційні мережі

У 1999 році Кевін Ештон назвав концепцію Інтернету речей та почав вивчати цю ідею, працюючи у виробничому гіганті Procter & Gamble як бренд-менеджер Oil of Olay. Він помітив розрив між системами запасів у роздрібних торговцях і фактичними запасами на полиці. Оскільки співробітники та менеджери магазинів сканували штрих-коди продуктів, щоб продати їх, вони припускали, що існує якесь внутрішнє відстеження, яке дозволить їм знати, коли товарів немає в наявності. Однак цього не було, доки Ештон не запровадила технологію радіочастотної ідентифікації (RFID) . Надавши реєстр для обліку продуктів та системи запасів, Ештон закрити розрив між відстеженням запасів та управлінням ланцюгом поставок.

Хоча Ештон ввів термін «Інтернет речей», пізніше він сказав, що назва «Інтернет для речей» могла б бути кращим описом.

Ештон був першим у використанні Інтернет речей, а пізніше він продовжував допомагати баченню стати відчутним в реальності як засновник Auto-ID Labs. Цей центр виріс у глобальну організацію на чотирьох континентах, яка проводить дослідження технологій RFID та інших датчиків, які живлять IoT.

Однак уявлення про те, що машини можуть спілкуватися один з одним, сягає 1800-х років. Інновації, такі як перший голосовий радіозв'язок у 1900 році,

були фундаментальними для розвитку Інтернету речей. Не менш важливими були й інші винаходи, такі як супутники, Інтернет та IP-адреса.

Одним з пристроїв, який заклав початок IoT у 1989 році, був торговий автомат Соса-Сола в Університеті Карнегі-Меллона. Студенти з факультету програмування з'ясували, як підключитися до торгового автомата через ARPANET (рання версія Інтернету) і побачити, чи є в наявності бажаний напій, перш ніж вони підйдуть до автомата. Однак, оскільки це все ще вимагало участі людини, машина Соса-Сола не зовсім відповідала повному визначенню пристрою IoT, який ми використовуємо сьогодні.

Це сталося в 1991 році, коли Джон Ромкі та Саймон Хакетт представили вдосконалену версію Інтернет-тостера, укомплектовану автоматизованим краном, який міг вставляти та виймати хліб для користувача. Навіть у ті часи тостер Ромкі та Хакетта відповідав деяким питанням безпеки та конфіденційності. Хоча пара новаторів усвідомлювала, що їхній проект трохи безглуздий, насправді це було частиною розваги, вони також усвідомлювали його серйозні наслідки для майбутнього технології Інтернет речей.

Перша веб-камера, створена для контролю кількості кави в кавнику, з'явилася в Кембриджі в 1993 році. Незабаром у 1994 році новатор Стів Манн підключив камеру до своїх окулярів і зміг вести пряму трансляцію в Інтернеті.

Ці ранні приклади пристроїв IoT також були основоположниками майбутніх соціальних медіа, смартфонів і навіть потокових передач.

Технологія RFID, започаткована Ештоном, отримала підтримку від таких великих організацій, як Міністерство оборони США та Walmart. До 2005 року IoT був достатньо великою угодою, ніж технологією.

Ця технологія, розвивається швидше, ніж багато людей це розуміють. Щоб зрозуміти, у 2018 році було 7 мільярдів пристроїв IoT. До 2019 року ця цифра зросла майже в чотири рази до 26,6 мільярдів. А в 2020 році до мережі було підключено 31 мільярд нових пристроїв Інтернету речей.

Інтернет речей містить в собі два терміни:

- інтернет
- речі

Інтернет - це мережа взаємопов'язаних комп'ютерів у всьому світі, які можуть спілкуватися один з одним.

Що стосується «речей» у рівнянні, то це пристрої та об'єкти, які можуть підключатися до мережі без участі людини. Останній пункт дуже важливий, оскільки це означає, що такі пристрої, як комп'ютери та смартфони, як правило, не вважаються безпосередньою частиною Інтернету речей. Хоча ваш телефон або комп'ютер може давати вам команди для пристроїв Інтернету речей у вашому домі чи офісі, він все ще залежить від ваших дій.

Якщо об'єднати ці два терміни, ми отримаємо термін Інтернет речей, що означає мережу фізичних пристроїв.

Незамінним елементом у цьому терміні є слово «підключений», оскільки система IoT існує лише тоді, коли її гаджети (або парк IoT) підключені через Інтернет. Якщо ми розірвемо зв'язок і розділимо всі гаджети, система IoT перестане існувати і втратить свої переваги.

Отже, говорячи про IoT, ми говоримо про зв'язність його речей. Чим більше речей підключено, тим міцніша мережа.

Кількість підключених пристроїв у всьому світі вражає – у 2019 році в глобальній мережі IoT було зареєстровано 10 мільярдів різних речей.

Однак ця технологія стосується не тільки пристроїв. Величезна екосистема IoT нараховує до 7 компонентів у своїй архітектурі.

1.2 Компоненти архітектури IoT

Основними компонентами Інтернету речей є:

- пристрої
- мережа

– хмара

Пам'ятаючи про них, ви можете створити просту екосистему Інтернету речей. Однак, чим складнішу систему ви хочете розробити, тим більше компонентів вам потрібно використовувати.

Якщо розглянути найпопулярнішу архітектуру IoT, ми знайдемо наступні 7 компонентів.

Компонент №1: Пристрої Інтернету речей

— є основним елементом будь-якої мережі. Це фізичні пристрої, які бездротово підключаються до мережі IoT і мають можливість передавати дані через Інтернет.

Компонент №2: Мережа

— є сполучною речовиною між усіма пристроями. Мережа представлена протоколами та методами зв'язку, які дозволяють гаджетам спілкуватися один з одним, передавати дані та підключатися до хмари.

Компонент №3: Безпека

— є життєво важливим компонентом, оскільки він керує доступом до мережі IoT поза системою, запобігає витоку даних та багато іншого. Компонент безпеки може мати вирішальне значення для великих промислових додатків, які використовують IoT у виробництві, стаючи таким чином залежним від надійної роботи всієї системи. Компанії, які виробляють масову продукцію для розумних будинків або особистого користування, також повинні приділяти особливу увагу цьому компоненту, оскільки конфіденційність їхніх клієнтів має бути головним пріоритетом.

Компонент №4: Шлюз

— можна сприймати буквально як ворота, через які потрібно пройти. Це фізичний пристрій або віртуальна платформа, яка з'єднує датчики, модулі Інтернету речей та розумні пристрої до хмари.

Компонент №5: Хмара

— це мозок IoT. Це стосується технологій хмарних обчислень, які зберігають дані та використовують різні методи для аналізу та керування даними. Хмара – це місце, де дані перетворюються з неструктурованого інформаційного безладу в приемний потік цінних ідей. Завдяки великим даним і машинному навчанню, хмара дає нам чудові явища, такі як прогнозне обслуговування, автоматизація та багато іншого.

Компонент №6: Додаток

— це графічний інтерфейс, який дозволяє користувачам, власникам бізнесу, співробітникам та іншим людям спілкуватися з системою IoT, керувати її парком, перевіряти статус тощо.

Компонент №7: Користувачі

— всі люди, які впливають на систему Інтернету речей і пожинають плоди її роботи.

За допомогою цих компонентів можна описати будь-яку систему IoT.

1.2.1 Архітектура IoT

Технологія Інтернету речей (IoT) має широкий спектр застосувань, і використання Інтернету речей зростає все швидше. Залежно від різних сфер застосування Інтернету речей, він працює відповідно до того, як його було розроблено/розроблено. Але він не має стандартно визначеної архітектури роботи, якої суворо дотримуються повсюдно. Архітектура IoT залежить від її функціональності та реалізації в різних секторах. Тим не менш, існує базовий потік процесу, на основі якого будується Інтернет речей.

На рисунку 1.1 зображено 4-етапну архітектуру IoT.

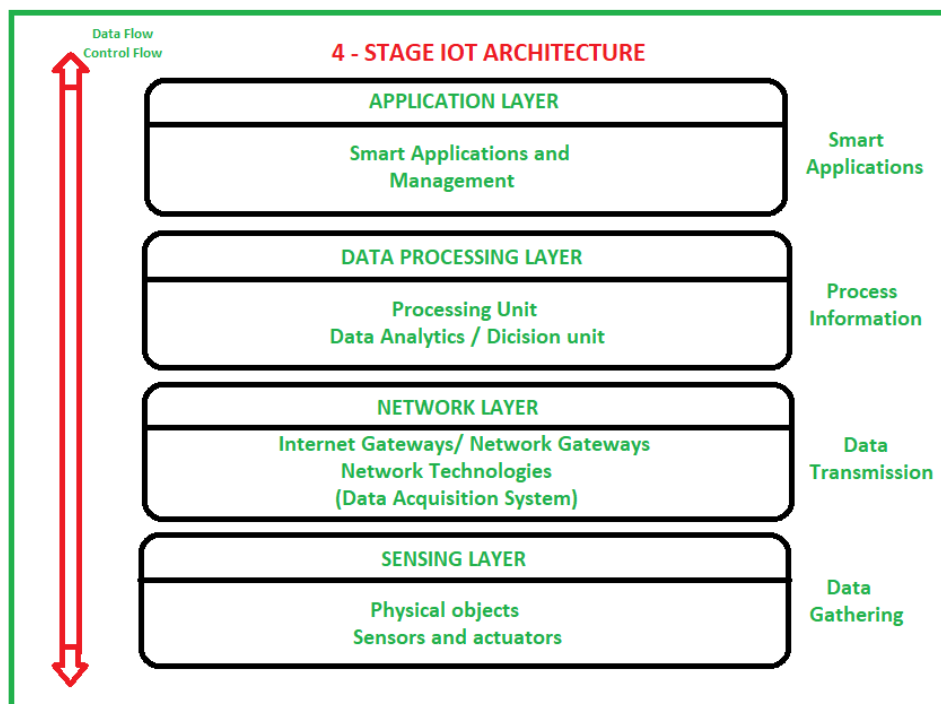


Рисунок 1.1 – етапна архітектура IoT

Отже, із зображення вище видно, що є 4 шари, які можна розділити наступним чином: Sensing Layer, Network Layer, Data Processing Layer і Application Layer.

– Чутливий рівень :

Датчики, виконавчі механізми, пристрої присутні в цьому шарі Sensing. Ці датчики або виконавчі механізми приймають дані (фізичні/екологічні параметри), обробляють дані та видають дані по мережі.

– Мережевий рівень :

На цьому рівні присутні шлюзи Інтернету/мережі, система збору даних (DAS). DAS виконує функцію агрегації та перетворення даних (збір даних і агрегування даних, а потім перетворення аналогових даних датчиків у цифрові дані тощо). Розширені шлюзи, які в основному відкривають з'єднання між мережами Sensor та Інтернетом, також виконують багато основних функцій шлюзу, як-от захист від шкідливого програмного забезпечення, а також

фільтрацію, а також прийняття рішень на основі введених даних і служб керування даними тощо.

– Рівень обробки даних :

Це процесорний блок екосистеми IoT. Тут дані аналізуються та попередньо обробляються перед відправкою в центр обробки даних, звідки до них звертаються програмні програми, які часто називають бізнес-додатками, де дані контролюються та керуються, а також готуються подальші дії. Отже, ось Edge IT або периферійна аналітика виходить на сцену.

– Рівень програми :

Це останній рівень з 4 етапів архітектури IoT. Центри обробки даних або хмара — це етап керування даними, на якому керують даними та використовуються додатками кінцевих користувачів, такими як сільське господарство, охорона здоров'я, аерокосмічна промисловість, сільське господарство, оборона тощо.

1.2.2 Програмне забезпечення IoT

Програмне забезпечення та мови програмування, на яких працює IoT, використовують дуже поширені мови програмування, які програмісти використовують і вже знають. То яку мову слід вибрати?

По-перше, оскільки вбудовані системи мають меншу потужність зберігання та обробки, їхні мовні потреби відрізняються. Найбільш часто використовуваними операційними системами для таких вбудованих систем є Linux або UNIX-подібні ОС, такі як Ubuntu Core або Android.

Програмне забезпечення Інтернету речей охоплює широкий спектр програмного забезпечення та мов програмування від мов загального призначення, таких як C++ і Java, до вбудованих специфічних варіантів, таких як мова Google Go або Parasail.

Ось короткий огляд кожного з програмного забезпечення IoT-

C & C++: мова програмування C має свої коріння у вбудованих системах — вона навіть почала програмувати телефонні комутатори. Він досить поширений, тобто його можна використовувати майже скрізь, і багато програмістів це вже знають. C++ – це об'єктно-орієнтована версія C, яка є популярною мовою як для ОС Linux, так і для вбудованих програмних систем IoT Arduino. Ці мови були в основному написані для апаратних систем, що робить їх такими простими у використанні.

Java: Хоча C і C++ специфічні для обладнання, код в JAVA є більш переносимим. Це більше схоже на мову писати один раз і читати будь-де, де ви встановлюєте бібліотеки, витрачаєте час на написання кодів один раз, і ви готові.

Python: останнім часом кількість користувачів Python різко зросла, і тепер він став однією з «відповідних» мов у веб-розробці. Його використання повільно поширюється на світ вбудованого керування та Інтернету речей, особливо на процесор Raspberry Pi. Python — це інтерпретована мова, яку легко читати, швидко вивчати та швидко писати. Крім того, це потужний пристрій для обслуговування додатків із великими обсягами даних.

В#: На відміну від більшості згаданих мов, В# був спеціально розроблений для вбудованих систем, він маленький і компактний і має менший розмір пам'яті.

Збір даних: використовується для фільтрації даних, захисту даних, зондування та вимірювання. Протоколи допомагають приймати рішення, відчувачи об'єкти в режимі реального часу. Він може працювати в обох напрямках, збираючи дані з пристроїв або розповсюджуючи дані на пристрої. Всі дані передаються на центральний сервер.

Інтеграція пристрою: це програмне забезпечення гарантує, що пристрої підключаються та підключаються до мереж, що полегшує обмін інформацією. Стабільна співпраця та зв'язок забезпечують між кількома пристроями.

Аналітика в режимі реального часу: при цьому вхідні дані від користувачів слугують потенційними даними для проведення аналізу в режимі реального часу, створення ідей, пропонування рекомендацій щодо вирішення проблем організації та покращення її підходу. Це, як наслідок, забезпечує автоматизацію та підвищення продуктивності.

Розширення додатків і процесів: ці програми розширюють охоплення існуючих систем і програмного забезпечення, щоб створити ширшу та ефективнішу систему. Вони інтегрують попередньо визначені пристрої для певних цілей, наприклад надання доступу певним мобільним пристроям або інженерним інструментам. Він підтримує підвищення продуктивності та більш точний збір даних.

1.3 Передача даних Інтернет речей (IoT)

Передача даних створених пристроєм, відбувається у три етапи. Перший етап - це початкове створення, яке відбувається на пристрої, а потім надсилається через Інтернет. Другий етап полягає в тому, як центральна система збирає та організовує ці дані. Третій етап – постійне використання цих даних у майбутньому.

Для розумних пристроїв і датчиків кожна подія може і буде створювати дані. Цю інформацію потім можна надіслати по мережі назад до центральної програми. На цьому етапі необхідно вирішити, за яким стандартом будуть створені дані та яким чином вони будуть надсилатися по мережі. Для повернення цих даних найчастіше використовуються стандартні протоколи MQTT, HTTP і CoAP. Кожен з них має свої переваги та варіанти використання.



Рисунок 1.2 – Мережі та протоколи IoT

HTTP забезпечує відповідний метод для передачі даних між пристроями та центральними системами. Спочатку розроблений для клієнт-серверної моделі обчислень, сьогодні він підтримує щоденний перегляд веб-сторінок до більш спеціалізованих служб на пристроях Інтернету речей. Хоча він відповідає функціональним вимогам для надсилання даних, HTTP містить набагато більше даних навколо повідомлення в його заголовках. Коли ви працюєте в умовах низької пропускної здатності, це може зробити HTTP менш придатним.

MQTT був розроблений як протокол для розгортання між машинами та Інтернетом речей. Він заснований на моделі публікації/підписки для доставки повідомлень з пристрою назад до центральної системи, яка діє як посередник, де вони потім можуть бути доставлені назад до всіх інших систем, які їх споживатимуть. Нові пристрої чи служби можуть просто підключатися до брокера, якщо їм потрібні повідомлення. MQTT легший за HTTP з точки зору розміру повідомлення, тому він корисніший для реалізацій, де пропускна здатність є потенційною проблемою. Однак він не включає шифрування як стандарт, тому це потрібно розглядати окремо.

Таблиця 1.1 – Порівняння протоколів MQTT та HTTP

1000 повідомлень	Кількість переданих байтів	Кількість пакетів	Час в секундах
MQTT	283743	265	5911
HTTP	15474263	12079	115669

Як можна побачити MQTT є в 20 разів швидшим та потребує в 50 разів менше трафіку в контексті відправки узгоджених даних, що залежать від певного часу

CoAP — це ще один стандарт, розроблений для середовищ з низьким енергоспоживанням і низькою пропускнуою здатністю. Замість того, щоб бути розробленим для брокерської системи, як MQTT, CoAP більше націлений на підключення один до одного. Він розроблений, щоб задовольнити вимоги дизайну REST, забезпечуючи спосіб інтерфейсу з HTTP, але все ще відповідає вимогам малопотужних пристроїв і середовищ.

Кожен з цих протоколів підтримує отримання інформації або оновлень з окремого пристрою та надсилання їх до центрального місця. Однак, де є більші можливості, так це те, як ці дані потім зберігаються та використовуються в майбутньому. Тут є дві основні проблеми: як дані діють, коли вони надходять у програму, і як вони зберігаються для подальшого використання.

Висновки

В данному розділі було розглянуто:

- Поява та розвиток технологій IoT;
- Основні компоненти Інформаційної системи та їх значення;
- Програмне забезпечення та мови програмування які використовуються в IoT;
- Передача даних та порівняння її двох найпопулярніших протоколів MQTT і HTTP.

2 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Компоненти системи IoT

Обладнання Інтернету речей включає широкий спектр пристроїв, таких як пристрої для маршрутизації, мости, датчики тощо. Ці пристрої Інтернету речей керують ключовими завданнями та функціями, такими як активація системи, безпека, специфікації дій, зв'язок та виявлення специфічних для підтримки цілей і дій.

Компоненти обладнання IoT можуть відрізнятися від плат малої потужності; одноплатні процесори, такі як Arduino Uno, які в основному є невеликими платами, які підключаються до материнської плати, щоб покращити та підвищити її функціональність, показуючи конкретні функції або функції (наприклад, GPS, датчики світла та тепла або інтерактивні дисплеї).

Програміст визначає вхід і вихід плати, потім створює схему, щоб проілюструвати взаємодію цих входів і виходів.



Рисунок 2.1 Arduino Uno

Іншою відомою платформою IoT є Raspberry Pi 2, який є дуже доступним і крихітним комп'ютером, який може включати в себе цілий веб-сервер. Його часто називають «RasPi», він має достатньо потужності обробки та пам'яті, щоб запустити на ньому Windows 10, а також IoT Core.

RasPi демонструє чудові можливості обробки, особливо при використанні мови програмування Python.

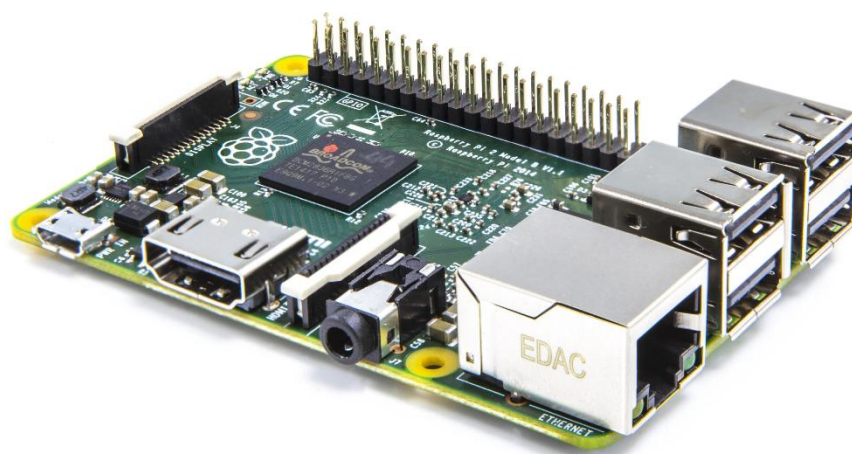


Рисунок 2.2 Raspberry Pi 2

BeagleBoard — це одноплатний комп'ютер з ОС на базі Linux, який використовує процесор ARM, здатний до більш потужної обробки, ніж RasPi.

Плати технологічного гіганта Intel Galileo та Edison — це інші варіанти, обидві відмінно підходять для великомасштабного виробництва, а Qualcomm виготовила низку технологій Інтернету речей на рівні підприємства для автомобілів та камер для охорони здоров'я.

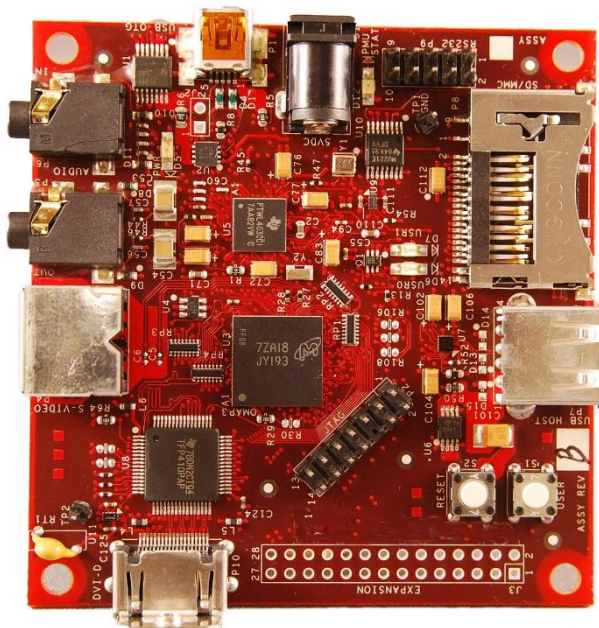


Рисунок 2.3 BeagleBoard

Найважливішим обладнанням IoT можуть бути його датчики. Ці пристрої складаються з різноманітних модулів, таких як енергетичні модулі, РЧ-модулі, модулі керування живленням і модулі датчиків.

Датчики та IoT забезпечують набагато кращу обізнаність про ситуацію, дозволяючи власникам активів одразу й віддалено знати речі, які без цих технологій було б дуже важко (або дуже дорого) знати взагалі. Це дозволяє приймати рішення на основі більш точних фактів.

За останні роки сенсорні технології різко змінилися. Датчики стали меншими, дешевшими та менш енергоємними, і ця тенденція прискорюється.

Датчики та IoT забезпечують набагато кращу обізнаність про ситуацію, дозволяючи власникам активів одразу й віддалено знати речі, які без цих

технологій було б дуже важко (або дуже дорого) знати взагалі. Це дозволяє приймати рішення на основі більш точних фактів.

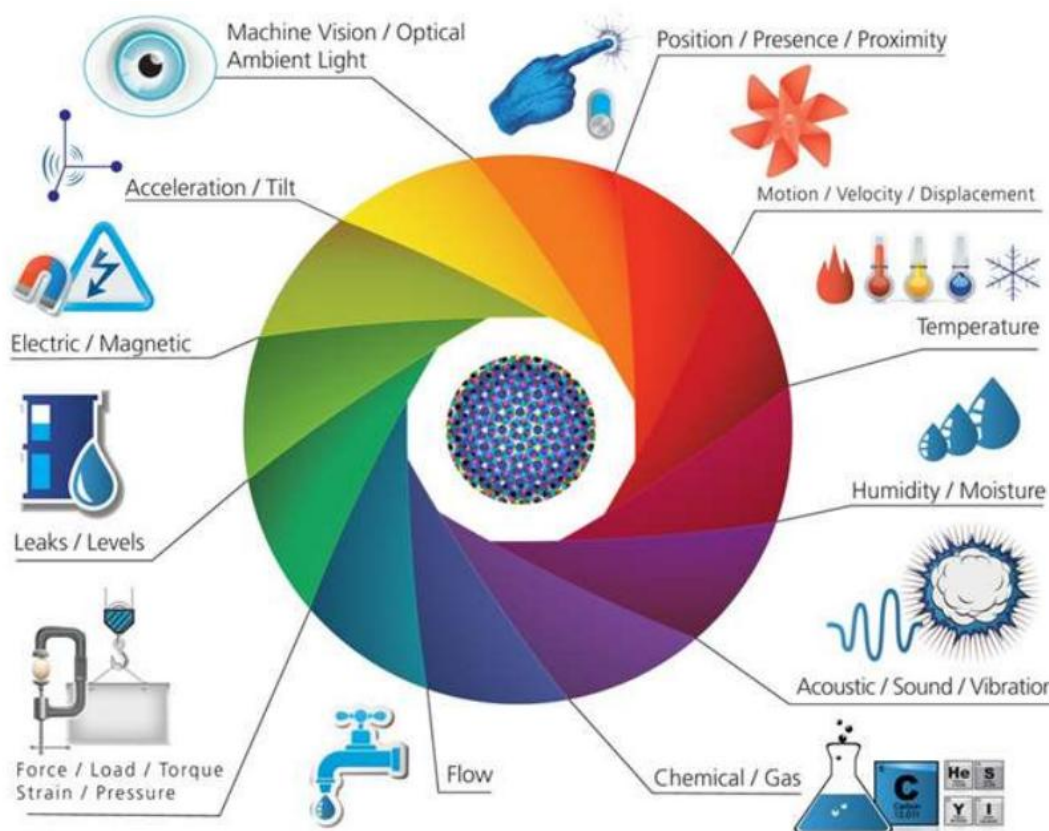


Рисунок 2.4 Різновиди датчиків

ІоТ надає потенціал для економії витрат і кращої обізнаності про ситуацію в активах. У міру розвитку комунікаційних технологій і мереж (а разом з ними і можливості передавати дані від величезної кількості датчиків), масове застосування датчиків усіх видів може бути корисним у багатьох сферах, про які раніше не думали. Сьогодні побутову техніку, як-от холодильники, можна підключити до Інтернету та надавати інформацію про їх вміст.

Сьогодні навіть звичайні смартфони містять такі датчики, як магнітометри, датчики наближення, барометри, гіроскопи та акселерометри. Поточними напрямками майбутньої розробки сенсорів є збір енергії для енергійно-автономних сенсорів, наносенсорів, біосенсорів та хімічних або екологічних датчиків. Масове застосування підключених датчиків у реальному часі також

полегшить нові способи проектування, експлуатації, обслуговування та перевірки активів, що сприятиме оптимізації протягом усього життєвого циклу.

Повсякденні пристрої, які ми використовуємо, такі як настільні комп'ютери, мобільні телефони та планшети, залишаються невід'ємною частиною системи IoT.

Робочий стіл надає користувачеві дуже високий рівень контролю над системою та її налаштуваннями.

Планшет виконує функцію пульта і забезпечує доступ до основних функцій системи.

Мобільний телефон дозволяє віддалено працювати та змінювати деякі важливі налаштування

Інші ключові підключені пристрої включають стандартні мережеві пристрої, такі як маршрутизатори та комутатори.

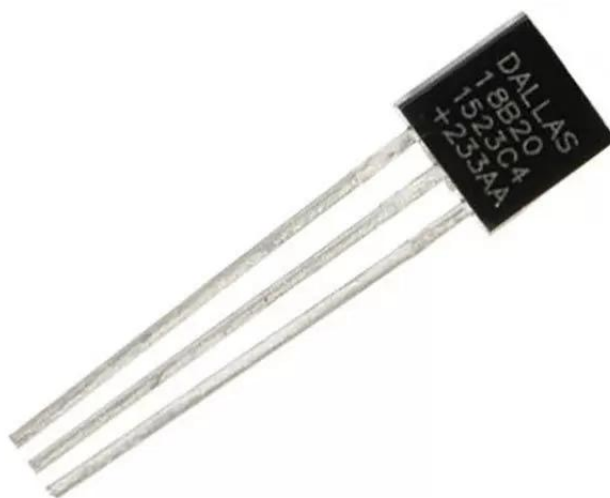


Рисунок 2.5 Датчик температури DS18B20

Цифровий датчик температури DS18B20 - це повноцінний цифровий термометр, здатний вимірювати температуру в діапазоні від -55°C до $+125^{\circ}\text{C}$ з програмованою точністю 9-12 біт. При виготовленні на виробництві, кожному датчику присвоюється власна унікальна 64-бітна адреса, а обмін інформацією з

провідним пристроєм (мікроконтролером або платою Arduino) здійснюється по шині 1-wire. Такий підхід дозволяє підключати до однієї лінії цілу групу датчиків, аж до 264.

Технічно датчики DS18B20 випускаються в 3-х корпусах: TO-92, SO, uSOP.



Рисунок 2.6 Датчик звуку Arduino

Датчик звуку Arduino використовується для стеження за рівнем шуму або виявлення гучних сигналів: хлопків, стукотів чи свисту. Датчик є невеликою платою з встановленим на ній мікрофоном, мікрофонним підсилювачем, регулятором чутливості у вигляді змінного резистора. Мікрофон перетворює звукові коливання на коливання електричного струму. Сигнал із мікрофона необхідно підсилити за допомогою компаратора L293. Датчик має вихід із логічним рівнем. Спрацював датчик - на виході з'явився логічний 0. Регулятором чутливості можна вибирати, від якого звуку спрацюватиме датчик - від слабого, гучного або дуже гучного звуку. Датчик має 3 висновки. Призначення висновків таке:

Vcc – живлення датчика;

GND – земля;

OUT – цифровий вихід;

Датчик має також світлодіод, що сигналізує про наявність низького рівня на виході OUT. Наявність цифрового виведення OUT та світлодіода рівня дозволяє використовувати модуль автономно, без підключення до контролера.

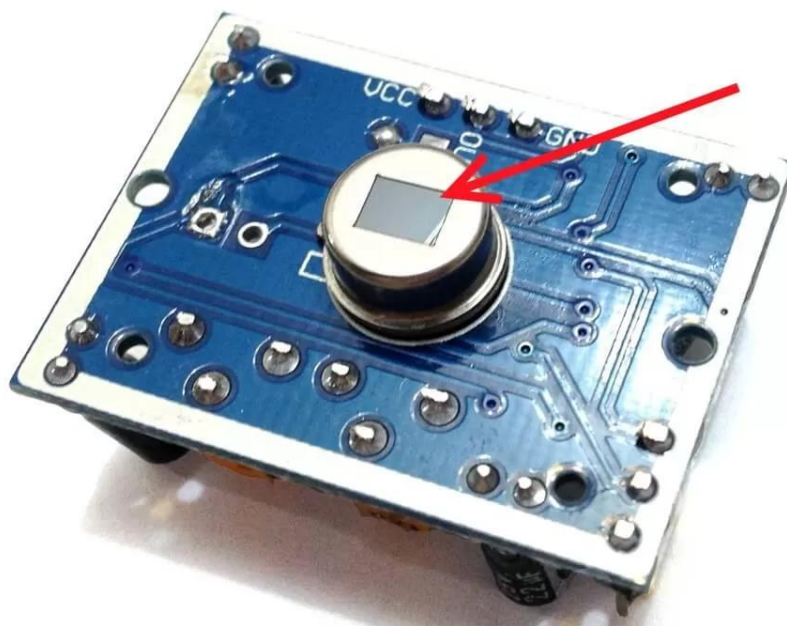


Рисунок 2.7 Датчик руху (або присутності) HCSR501

Модуль датчика руху (або присутності) HCSR501 на основі піроелектричного ефекту складається з PIR-датчика 500BP (рис. 1) з додатковою електричною розв'язкою на мікросхемі BISS0001 та лінзи Френеля, яка використовується для збільшення радіусу огляду та посилення інфрачервоного сигналу (рис. 1). Модуль використовується для виявлення руху об'єктів, що випромінюють інфрачервоне випромінювання. Чутливий елемент модуля – PIR-датчик 500BP. Принцип його роботи заснований на піроелектриці. Це виникнення електричного поля в кристалах при зміні їх температури.

Управління роботою датчика здійснює мікросхема BISS0001. На платі розташовано два потенціометри, за допомогою першого налаштовується дистанція виявлення об'єктів (від 3 до 7 м), за допомогою другого – затримка

після першого спрацьовування датчика (5 – 300 сек). Модуль має два режими – L та H. Режим роботи встановлюється за допомогою перемички. Режим L – режим одиничного спрацьовування, при виявленні об'єкта, що рухається, на виході OUT встановлюється високий рівень сигналу на час затримки, встановлений другим потенціометром. На цей час датчик не реагує на об'єкти, що рухаються. Цей режим можна використовувати в системах охорони для сигналізації тривоги на сирену. У режимі H датчик спрацьовує щоразу при виявленні руху. Цей режим можна використовувати для увімкнення освітлення. При включенні модуля відбувається його калібрування, тривалість калібрування приблизно одна хвилина після чого модуль готовий до роботи. Встановлювати датчик бажано далеко від відкритих джерел світла.

2.2 Загальний вигляд системи

Для демонстрації розробленої системи було вибрано приміщення однокімнатного дому яке включає в себе спальню, ванну кімнату, кухню студію і терасу.

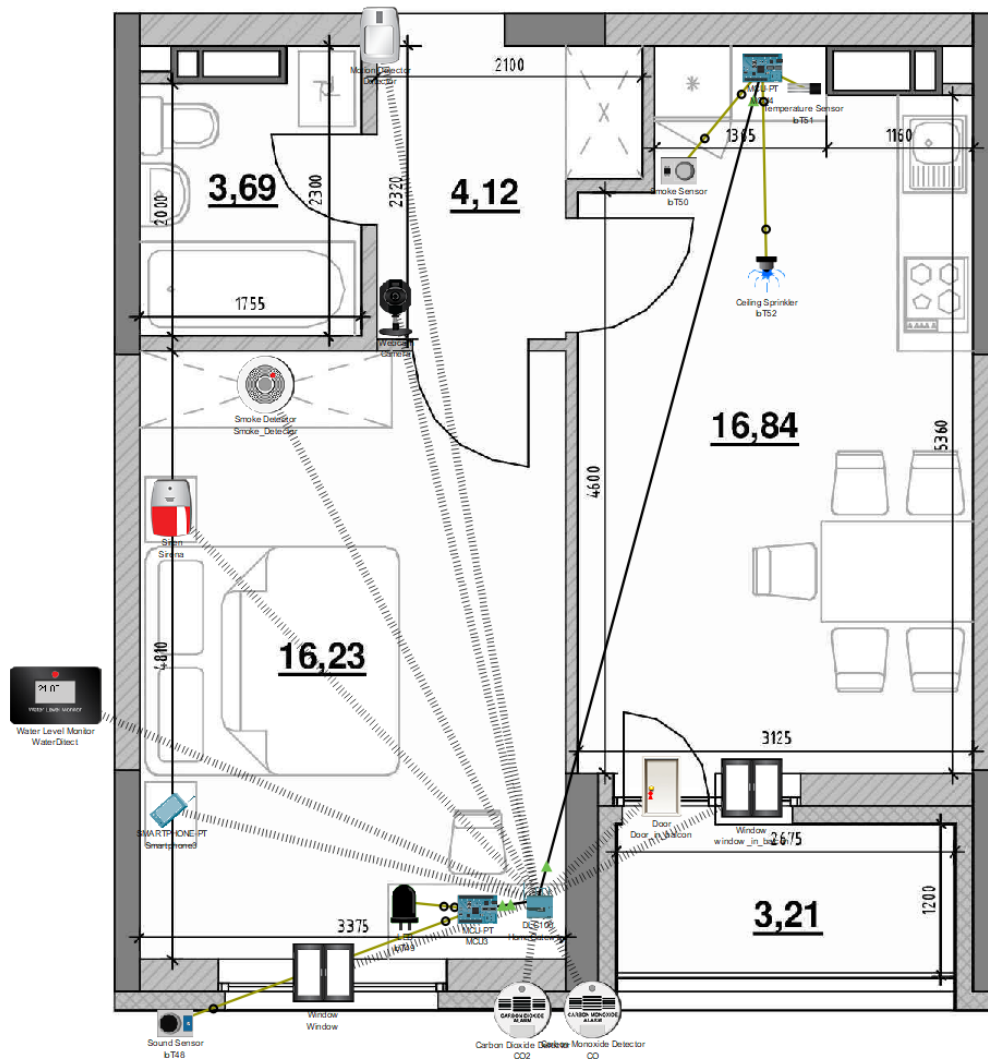


Рисунок 2.8 Загальна схема будинку з розташуванням датчиків і IoT пристроїв.

Датчик руху і камера будуть розташовані при вході в дім для того щоб фіксувати всіх хто заходить та виходить з дому щоб протидіяти незаконному проникненню в дім. На кухні будуть розташовані датчики диму і температури та пристрій гасіння пожежі який за певних умов які прописані в кодї керуванні датчиків буде вмикатися і запобігати вогнебезпечним ситуаціям. Також в спальні буде розташований пристрій який вловлює дим і буде за допомогою сирени повідомляти власника та відчиняти вікно у спальні. Всі інші датчики будуть розташовані зовні. Пристрої будуть уловлювати такі данні як рівень води, щоб повідомити про повінь, вмість монооксиду вуглецю та інших

шкідливих домішок, щоб закрити вікна і двері та перекрити доступ забрудненого повітря до будинку.

2.3. Налаштування мережевих параметрів

По-перше кожному пристрою потрібно призначити його власну ір-фдресу для було використано протокол DHCP.

DHCP — протокол прикладного рівня моделі TCP/IP, служить призначення IP-адреси клієнту. Це впливає з назви — Dynamic Host Configuration Protocol. IP-адресу можна призначати вручну кожному клієнту, тобто комп'ютеручи пристрою локальної мережі. Але у великих мережах це дуже трудомістка справа, до того ж, чим більше локальна мережа, тим більше зростає можливість помилки при налаштуванні. Тому для автоматизації призначення IP було вибрано протокол DHCP. Для роботи в мережі клієнту потрібно не тільки IP, але й інші параметри DHCP - наприклад, маска підмережі, стандартний шлюз і адреса сервера. Опції є пронумерованими пунктами, рядками даних, які містять необхідні клієнту сервера параметри конфігурації. Дамо опис деяким опціям:

- Option 1 - маска підмережі IP;
- Option 3 - основний шлюз;
- Option 6 - адреса сервера DNS (основна та резервна);
- Option 51 визначає, на який термін IP-адреса надається в оренду клієнту;
- Option 55 - список опцій, що запитуються. Клієнт завжди запитує опції для правильної конфігурації. Відправляючи повідомлення з Option 55, клієнт виставляє список числових кодів опцій, що запитуються, в порядку переваги. DHCP-сервер намагається надіслати відповідь з опціями в тому самому порядку.

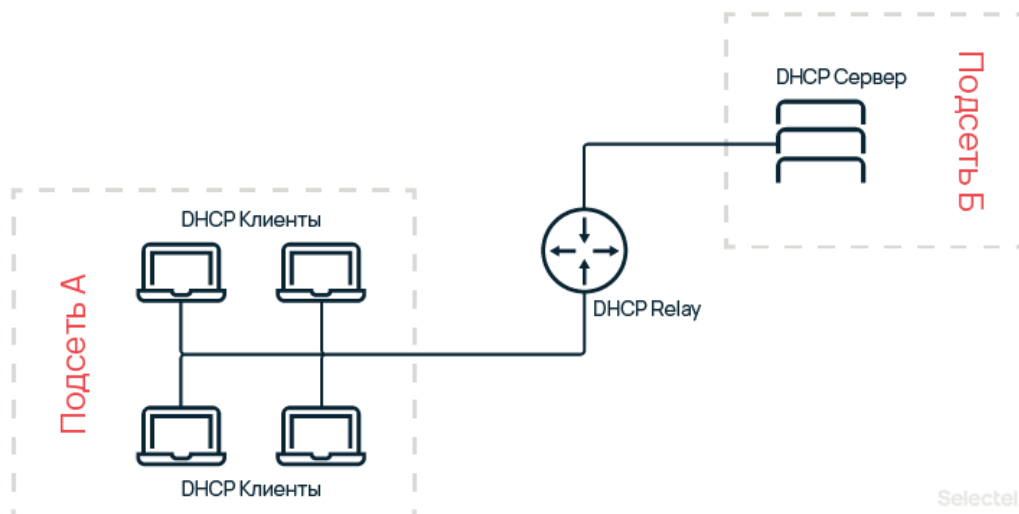


Рисунок 2.9 Работа протоколу DHCP

Для об'єднання всіх пристроїв та надсилання даних на смартфон було використано шлюз DLC100. Нижче показана його конфігурація.

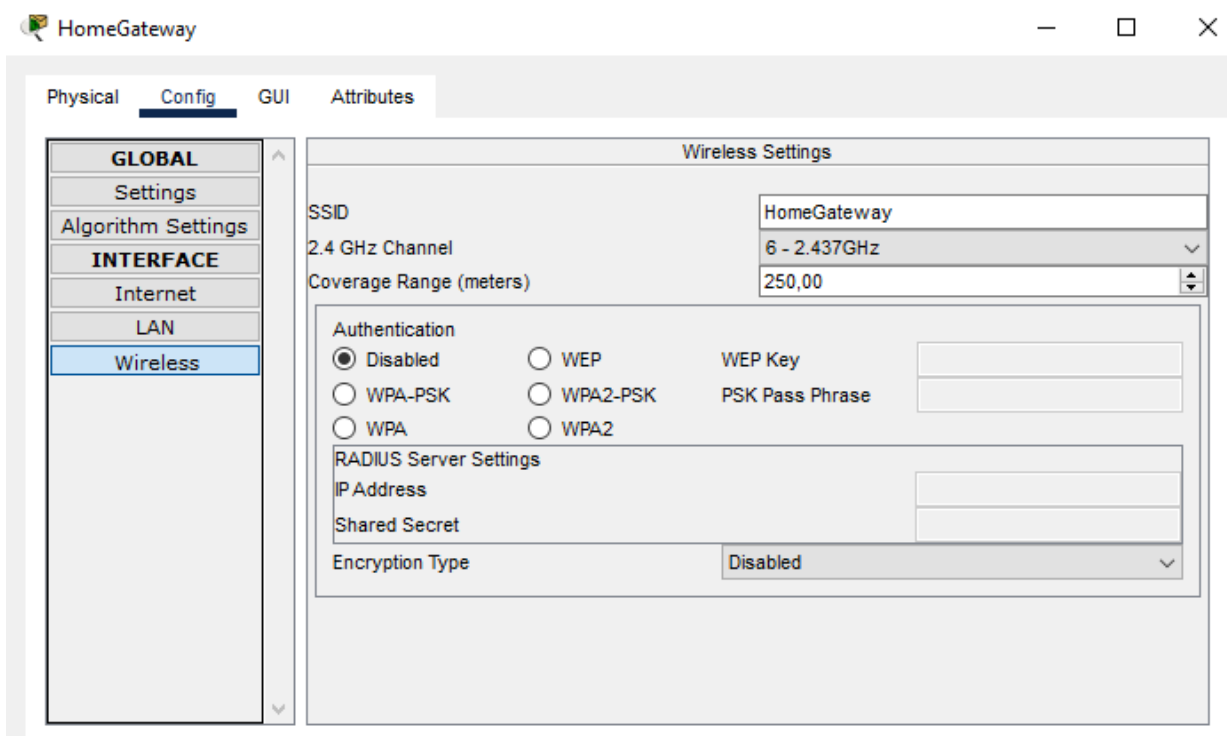


Рисунок 2.10 Налаштування домашнього шлюзу

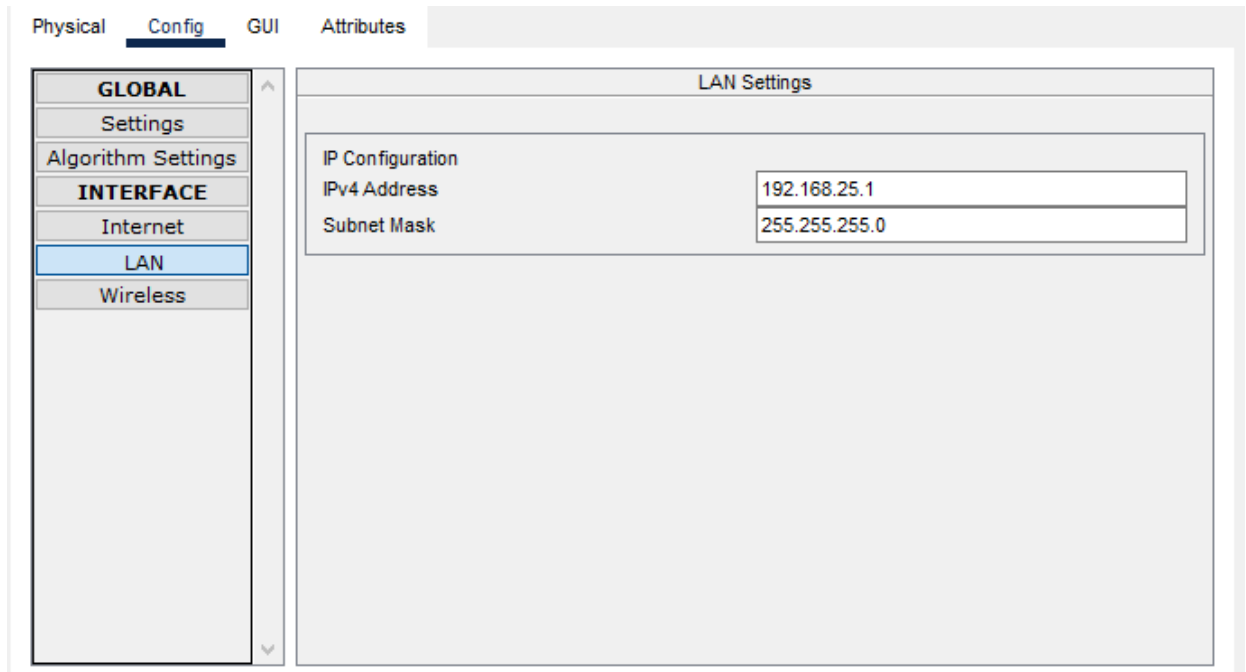


Рисунок 2.11 LAN налаштування шлюзу

Для підключення всіх пристроїв до шлюзу було використано модуль безпроводного доступу RT-IOT-NM-1W. Модуль RT-IOT-NM-1W забезпечує один бездротовий інтерфейс 2,4 ГГц, який підходить для підключення до бездротових мереж або домашнього шлюзу.

Щоб отримати доступ до всіх даних та пристроїв IoT було використано Iot monitor.

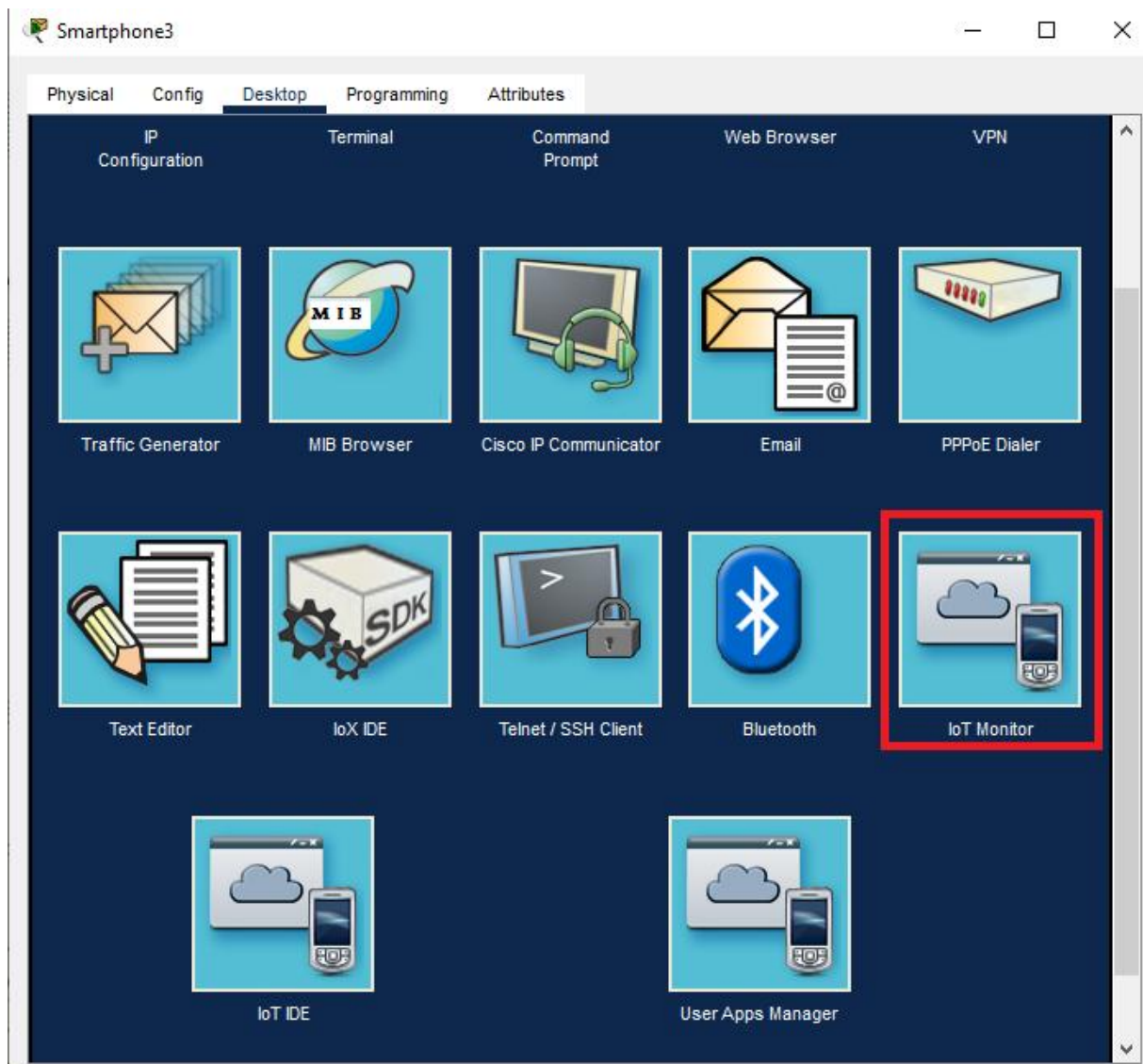


Рисунок 2.12 IoT Monitor

В ньому видно всі підключені пристрої і також можна прописати правила за якими вони працюють. Для того щоб увійти в систему нам потрібно ввести IP-адрес шлюзу та логін і пароль за замовчкванням це admin.

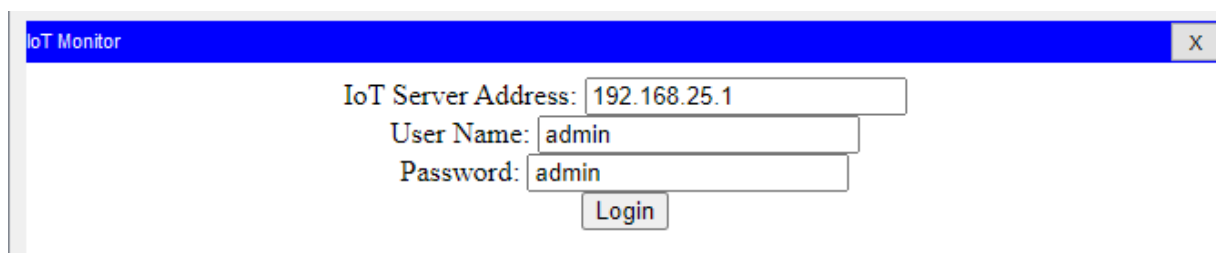


Рисунок 2.13 Вхід в IoT Monitor

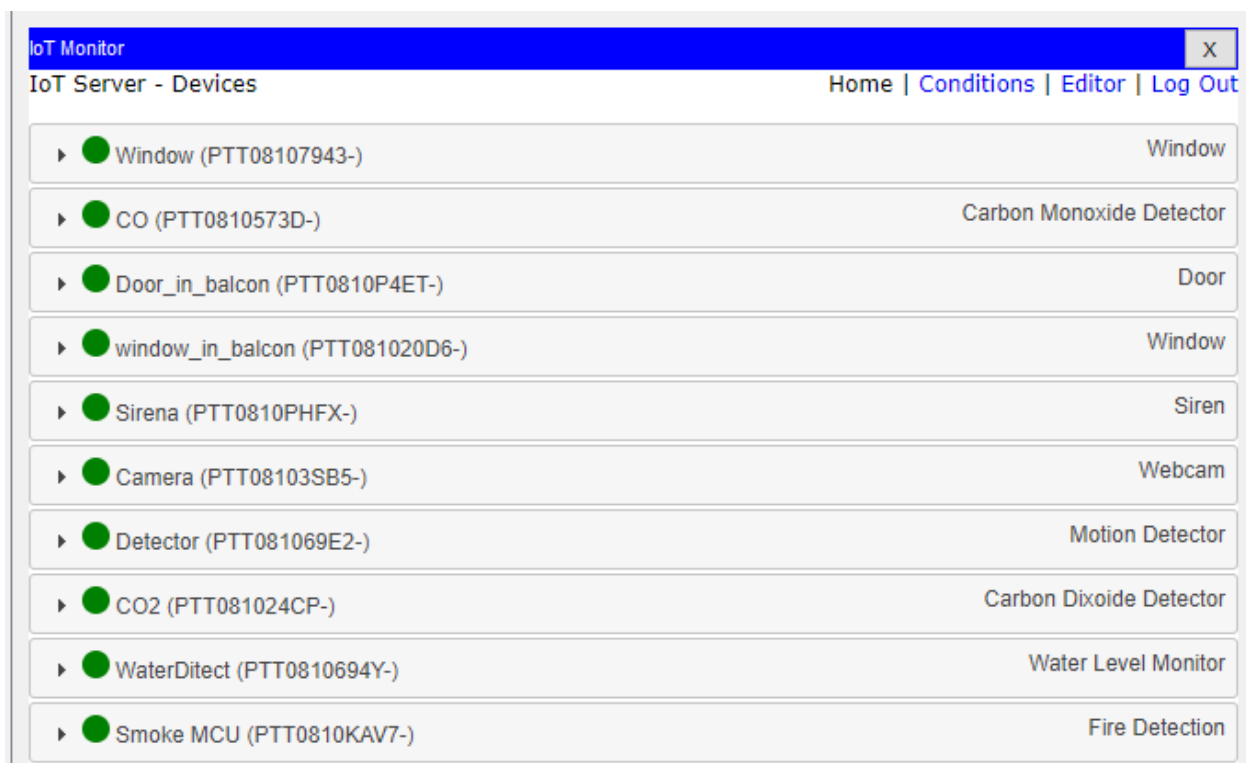


Рисунок 2.14 Девайси підключені до IoT Monitor

Висновки

В даному розділі були:

- Описані компоненти IoT системи, такі як плати і визначено що за основу для даного проєкту краще використати плати Arduino Uno;
- Представлені датчики які будуть використовуватись в даній інформаційній системі;
- Змодельовано загальний вигляд системи та показано розташування пристроїв і датчиків в залежності від плану будинку;
- Проведені налаштування та підключення пристроїв та датчиків до домашнього шлюзу.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Апаратна частина інформаційної системи

Метою апаратного забезпечення є збір даних і можливість контролювати необхідні функції. Основним обладнанням, яке використовується, є Arduino Uno з датчиками.

Основним обладнанням для цього проекту є Arduino Uno

Причина вибору полягала в тому, що

- Можна жити, програмувати та обмінюватися повідомленнями з Arduino за допомогою одного кабелю USB (або FTDI кабелю для деяких клонів).
- Можливо зробити простий проект за кілька хвилин, використовуючи стандартні бібліотеки, не вникаючи у них. Для зчитування сигналів кнопок, виведення інформації на семи сегментні або РК-дисплеї та управління двигунами для цього є стандартні бібліотеки, що не потребують великого досвіду в програмуванні.
- Послідовні та SPI інтерфейси зв'язку зроблені чудово

Використовуються датчики: камера, вікна, двері, блок детектування руху, датчик води, датчик газу, датчик звуку, датчик CO₂ s CO, сирена, датчик температури. Деякі датчики працюватимуть разом для досягнення мети, а інші працюватимуть самостійно. Наприклад, детектор руху дозволить камері знімати відео або знімки, коли він активується. Датчик води та газу будуть працювати окремо, щоб надіслати сповіщення у разі зміни будь-якого значення.

Щоб контролювати, куди і коли надсилається інформація з датчиків, є смартфон. Метою смартфона є передача інформації між пристроями підключеними до мережі IoT. Смартфон повинен мати можливість безпечно та ефективно підключатися до різноманітних датчиків і мобільних пристроїв.

Шлюз – це один із типів маршрутизаторів. Маршрутизатори з'єднують кілька мереж та виконують функції маршрутизації пакетів. Наприклад, деякі маршрутизатори передають дані по маршруту на рівні мережного інтерфейсу або фізично. Шлюзи здійснюють маршрутизацію на мережному рівні. При пересиланні пакетів шлюзи орієнтуються на адресу цільової мережі, а чи не на адресу конкретного хоста. Тобто, шлюз не повинен зберігати список маршрутів до всіх можливих цільових хостів пакета.

3.2. Інструменти розробки

Для розробки інформаційної системи був використаний Cisco Packet Tracer. Починаючи з 7 версії програми Cisco Packet Tracer у програмі з'явилась підтримка Internet of Things (Інтернет речей). Інтернет речей - це концепція мережі, яка складається з пристроїв, які здійснюють обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку. Отже розглянемо, які можливості надає нам Cisco Packet Tracer.

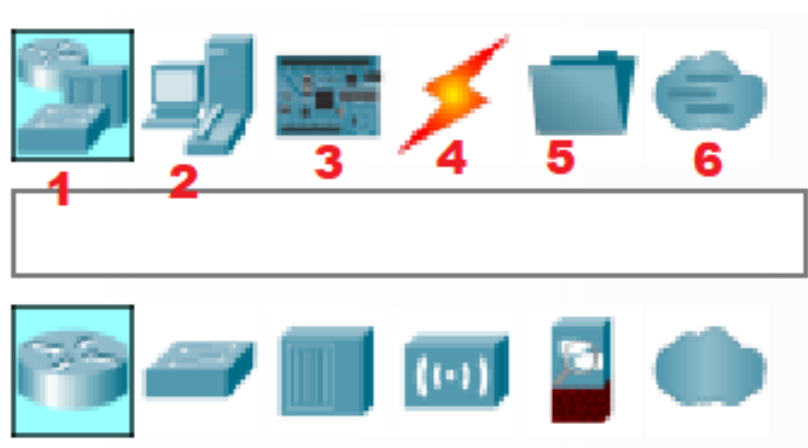


Рисунок 3.1 Панель управління

1 - мережеві пристрої

2 - кінцеві пристрої

3 - компоненти

4 - з'єднання

5 - різне

6 - з'єднання мультикористувача

Елементи Internet of Things знаходяться переважно в 2 вкладці. Проте є один елемент з вкладки 1. Цей елемент називається домашнім шлюзом.



Рисунок 3.2 Вкладка мережевих пристроїв(підгрупа бездротових пристроїв)

Жовтим маркером виділенні елементи, що використовуються в Internet of Things.

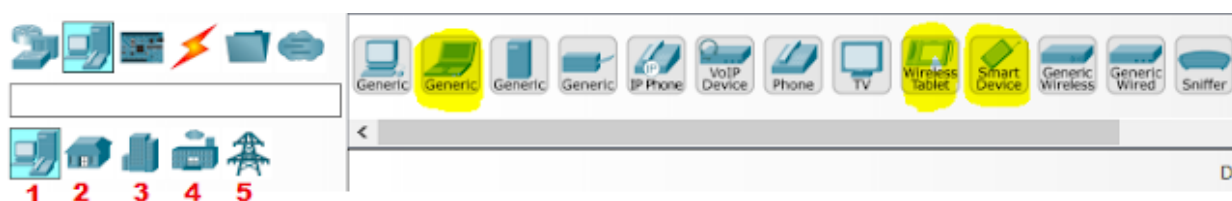


Рисунок 3.3 Вкладка кінцевих пристроїв

Кінцеві пристрої поділяються на групи:

кінцеві пристрої

дім, який містить такі елементи:



Рисунок 3.4 Компоненти вкладки "Дім"

В наступному списку всі компоненти перераховані так як вони зображені на малюнку: розумний кондиціонер, розумну кавоварку, батарею(акумулятор), блютуз динамік, детектори карбону діо- та моноксида, розумний вентилятор на стелі, розумні двері, розумний нагрівач, розумні двері гаражу, домашній динамік, розумний зволожувач, контролери зволоження, розумний поливач

газону, розумну лампу, детектор руху, портативний музичний програвач, вимірювач напруги, детектор диму, сонячну панель, вимірювач частоти звуку, контролер температури, розумний термостат, розумний водостікач.

3. розумне місто, яке містить:



Рисунок 3.5 Компоненти вкладки "Розумне місто"

- 3.1 контролер атмосферного тиску;
- 3.2 батарею (акумулятор);
- 3.3 передавач блютуз-маяків;
- 3.4 детектор карбон монооксиду;
- 3.5 розумний вентилятор на стелі;
- 3.6 стару машину;
- 3.7 вимірювач напруги;
- 3.8 картку радіочастотної ідентифікації;
- 3.9 зчитувач карток радіоактивної ідентифікації;
- 3.10 розумну LED лампу;
- 3.11 сонячну панель;
- 3.12 розумний вуличний ліхтар;
- 3.13 детектор вітру.

4. індустріальне, яке містить елементи:



Рисунок 3.6 Компоненти вкладки "Індустріальне"

- 4.1 розумний кондиціонер;
- 4.2 батарею (акумулятор);
- 4.3 детектор кабону моно- та діоксиду;
- 4.4 контролер вогню;
- 4.5 пожежогасник;
- 4.6 розумний нагрівач;
- 4.7 розумний зволожувач;
- 4.8 контролер зволоження;
- 4.9 вимірювач напруги;
- 4.10 картка радіочастотної ідентифікації;
- 4.11 зчитувач картки радіочастотної ідентифікації;
- 4.12 генератор сигналів;
- 4.13 розумну LED лампу;
- 4.14 сонячну панель;
- 4.15 контролер температури;
- 4.16 термостат;
- 4.17 детектор руху;
- 4.18 розумний водостікач;
- 4.19 детектор вітру;
- 4.20 вітрогенератор.

5. енергосистема, яка містить:



Рисунок 3.7 Компоненти вкладки "Енергосистема"

- 5.1 батарею(акумулятор);

- 5.2 повітродувку;
- 5.3 вимірювач напруги;
- 5.4 сонячну панель;
- 5.5 детектор вітру;
- 5.6 вітрогенератор;

Також ще використовуються елементи з 3 вкладки(компоненти). Всі компоненти можна програмувати відповідно до поставленого завдання або для досягнення бажаного результату. Ці компоненти можна застосовувати у проектах проектування розумних будинків, автоматизованих приміщень, розумних міст та інших проектів. Головною перевагою розумних речей є самостійне прийняття рішень та віддалене керування.



Рисунок 3.8 Вкладка "Компоненти"

1. Перша категорія - це плати. Тут присутні плати MCU та SBC, а також розумний пристрій. За допомогою цих плат (мікроконтролери та однопроцесорні плати) ми можемо програмувати речі (одну або декілька речей одночасно) та додавати їм новий функціонал. За допомогою елемента "пристрій" ми маємо можливість створити свою розумну річ, надати їй бажані стани, запрограмувати відповідним чином, обрати зображення для нашого пристрою, написати специфікацію пристрою, задати відповідні слоти та мережеві адаптери.

2. Друга категорія - це виконавчі механізми, які включають у себе:



Рисунок 3.9 Виконавчі механізми

- 2.1 охолоджувач повітря;
- 2.2 лампу тривоги;
- 2.3 розбризкувач на стелі;
- 2.4 затемнену лампу;
- 2.5 розбризкувач на підлозі;
- 2.6 нагрівальний елемент;
- 2.7 LCD;
- 2.8 LED;
- 2.9 мотор;
- 2.10 п'єзо динамік;
- 2.11 кольорову LED;
- 2.12 серводвигун;
- 2.13 розумну LED;
- 2.14 динамік.

3. І третя категорія - сенсори, містить:



Рисунок 3.10 Сенсори

- 3.1 сенсор гнучкості;
- 3.2 сенсор навколишнього середовища;
- 3.3 сенсор, який можна створити і запрограмувати;
- 3.4 сенсори вологи;
- 3.5 сенсор дискомфорту (вологість+температура)
- 3.6 мембранний потенціометр;
- 3.7 сенсор металу;
- 3.8 сенсор руху;

- 3.9 фото (світловий) сенсор;
- 3.10 потенціометр;
- 3.11 кнопку перемикання;
- 3.12 кнопку з одноразовим натиском;
- 3.13 вмикач;
- 3.14 сенсор диму;
- 3.15 сенсор звуку;
- 3.16 сенсор температури;
- 3.17 кнопку з постійним натиском;
- 3.18 лазерний сенсор руху;
- 3.19 детектор води;
- 3.20 сенсор води;
- 3.21 сенсор вітру.

3.3 Дослідження інформаційної системи

Для того щоб дослідити працездатність інформаційної системи потрібно налаштувати всі окремі пристрої та розробити скрипти для роботи датчиків.

Більшість уже готових IoT пристроїв налаштовуються однаково тому розглянемо приклад налаштування та підключення камери до загальної системи. Спочатку нам потрібно натиснути на камеру вибрати в IoT server назву нашого шлюзу, а потім вибрати advanced.

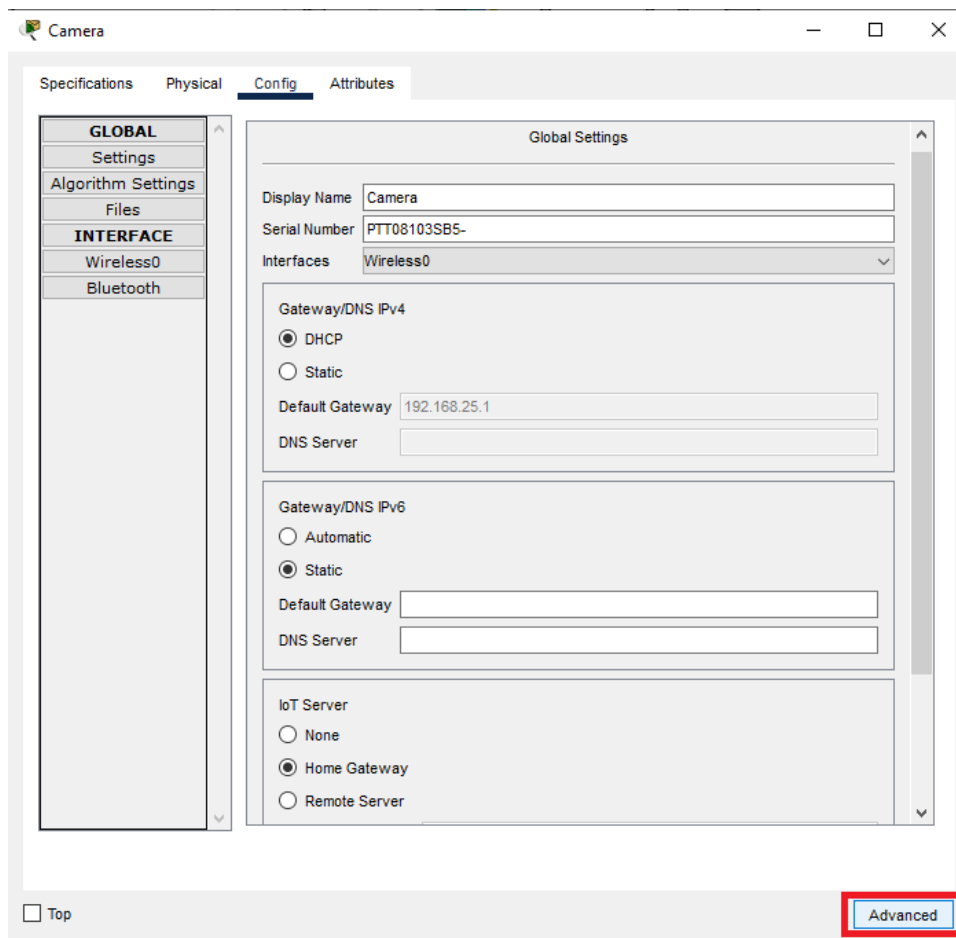


Рисунок 3.11 Вкладка Advanced

Потім перейти у вкладку I/O config та вибрати спосіб з'єднання з шлюзом.

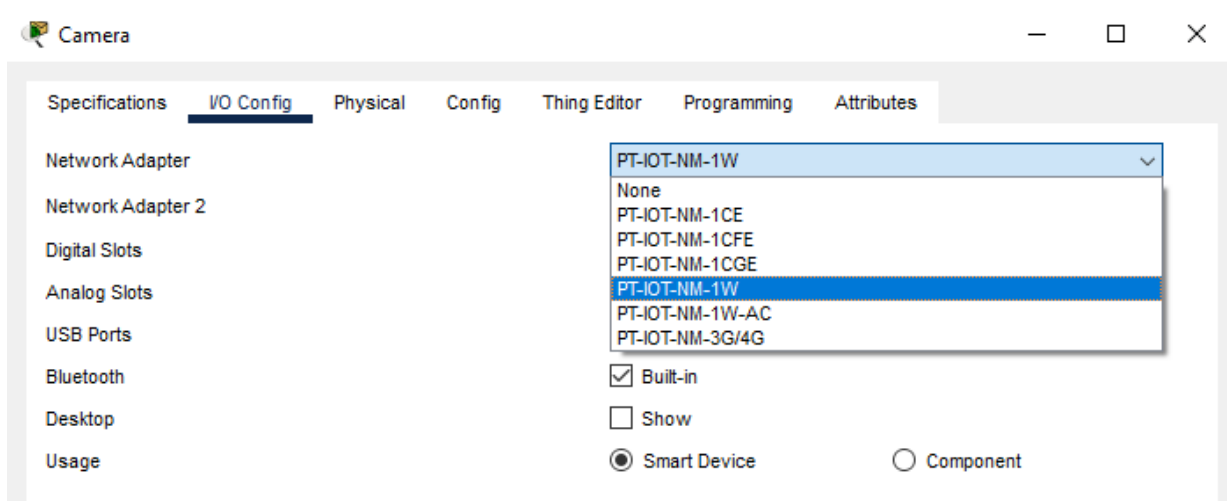


Рисунок 3.12 Вкладка I/O config

Підключення всіх інших девайсів відбувається аналогічно.

Для підключення окремих датчиків використовуємо плату MCU. Підключаємо всі датчики до відповідних пінів і загрузаємо на плату скрипт який буде запускати сирену і систему гасіння пожежі коли рівень диму буде більше 50 а температура вища 580 і запускаємо.

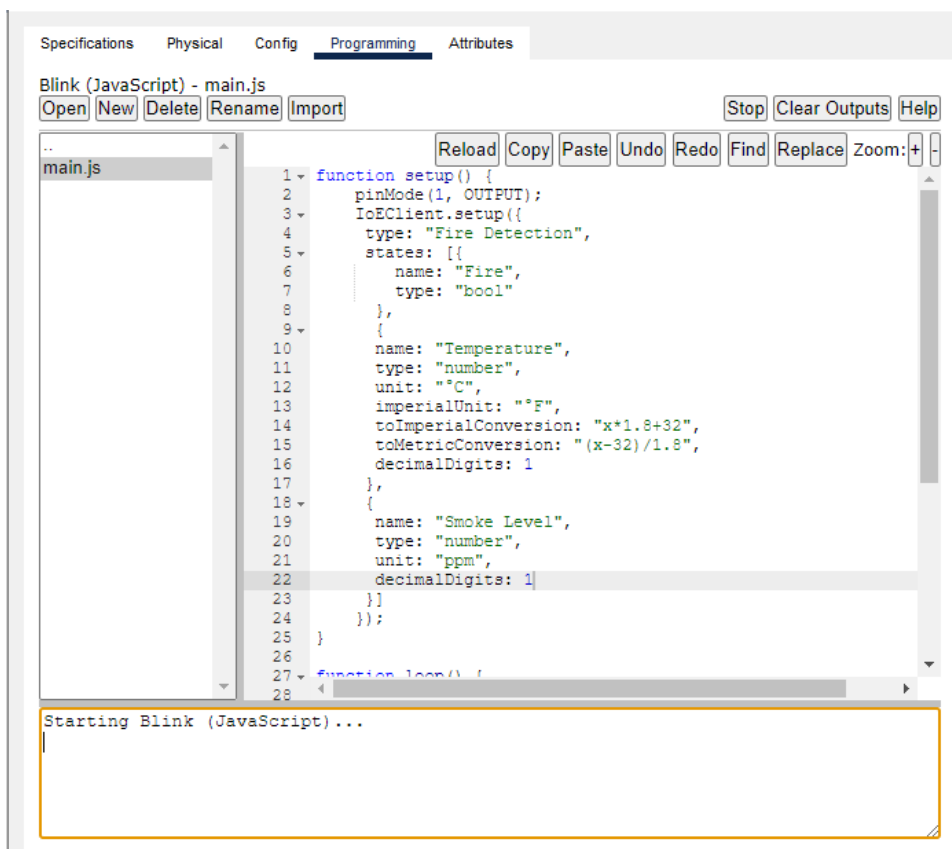


Рисунок 3.13 Програмування плати MCU

Нижче наведено сам скрипт:

```

function setup() {
pinMode(1, OUTPUT);
IoEClient.setup({
  type: "Fire Detection",
  states: [{
name: "Fire",
type: "bool"
  }],
  {

```

```
name: "Temperature",
type: "number",
unit: "°C",
imperialUnit: "°F",
toImperialConversion: "x*1.8+32",
toMetricConversion: "(x-32)/1.8",
decimalDigits: 1
},
{
name: "Smoke Level",
type: "number",
unit: "ppm",
decimalDigits: 1
}]
});
}

function loop() {
var SmokeLevel = analogRead(A0);
var Temperature = analogRead(A1);
var FireDetected=false;

if (SmokeLevel>50 || Temperature>580) {
    digitalWrite(0, HIGH);
    FireDetected=true;
}

else {
    digitalWrite(0, LOW);
    FireDetected=false;
}
```

```

}
IoEClient.reportStates([FireDetected, Temperature, SmokeLevel]);
delay(500);
}

```

3.4 Тестування системи

Для тестування системи ми з імітуємо різні погодні явища та ситуації змінюючи дані в симуляції. Внесені зміни можна побачити на графіку.

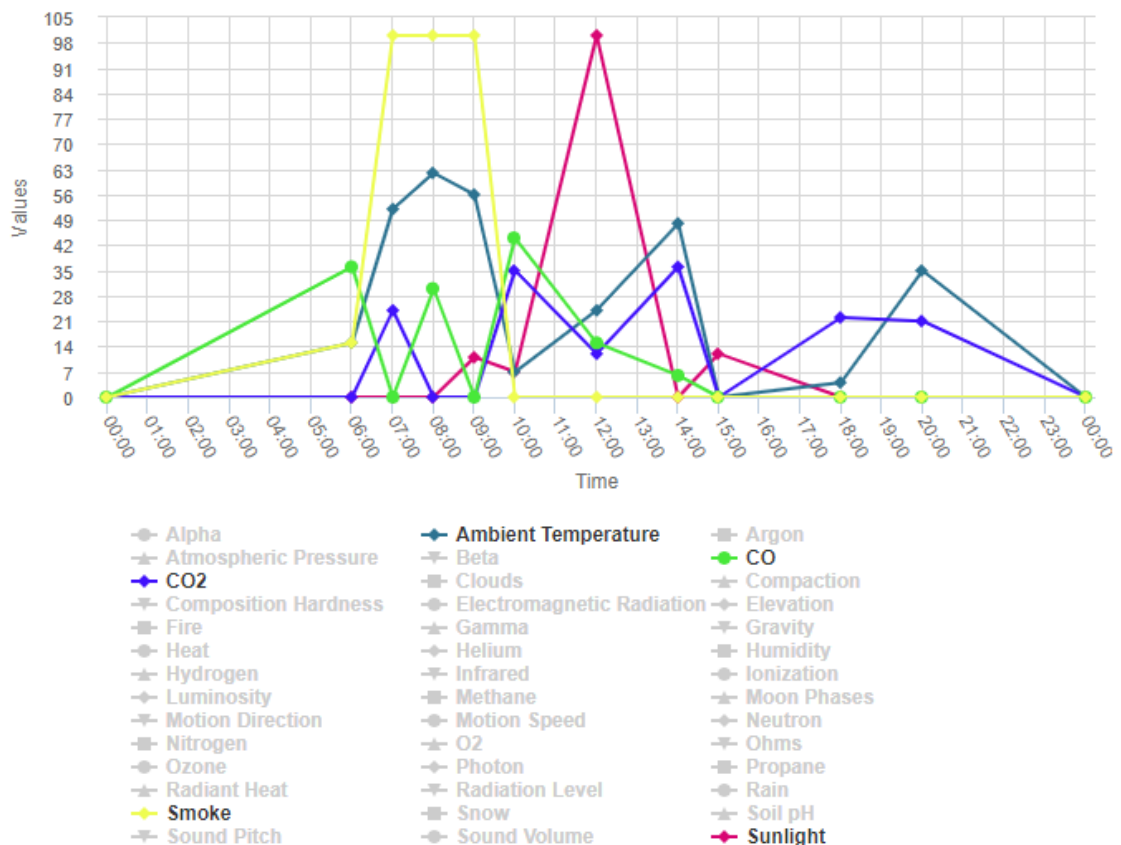


Рисунок 3.14 Графік залежності параметрів від часу

Для того щоб пристрої розуміли що і коли потрібно робити пропишемо правила в програмі IoT monitor.

IoT Monitor				
IoT Server - Device Conditions			Home Conditions Editor Log Out	
Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	Motion_detect	Detector On is true	Set Camera On to true Set Sirena On to true
Edit Remove	Yes	CO2 ditect	CO2 Alarm is true	Set Window On to false Set window_in_balcon On to false Set Door_in_balcon Lock to Lock
Edit Remove	Yes	WaterDisaster	WaterDitect Water Level >= 20.0 cm	Set Sirena On to true
Edit Remove	Yes	CO ditect	CO Alarm is true	Set Sirena On to true Set Window On to false Set window_in_balcon On to false Set Door_in_balcon Lock to Lock
Edit Remove	Yes	Water normal	WaterDitect Water Level <= 19.0 cm	Set Sirena On to false
Edit Remove	Yes	Motion_no_detect	Detector On is false	Set Camera On to false Set Sirena On to false
Edit Remove	Yes	Smoking	PTT081020EF- Alarm is true	Set Sirena On to true Set Window On to true

Рисунок 3.15 Правила роботи датчиків

Нижче будуть показані тести і їх результати :

- Коли датчик диму засік перевищення норми то він дав сигнал сирені і відкрив вікно у спальні.

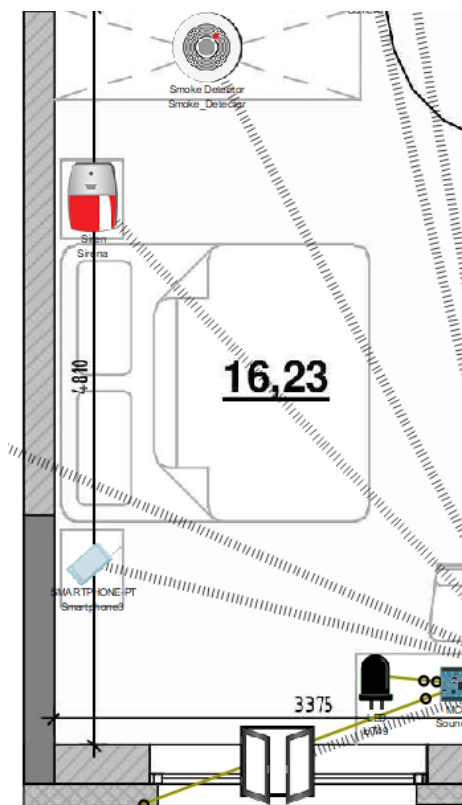


Рисунок 3.16 Тест 1

- Коли датчик рівня води засік перевищення заданої норми 20см (повінь) то він дав сигнал сирені.

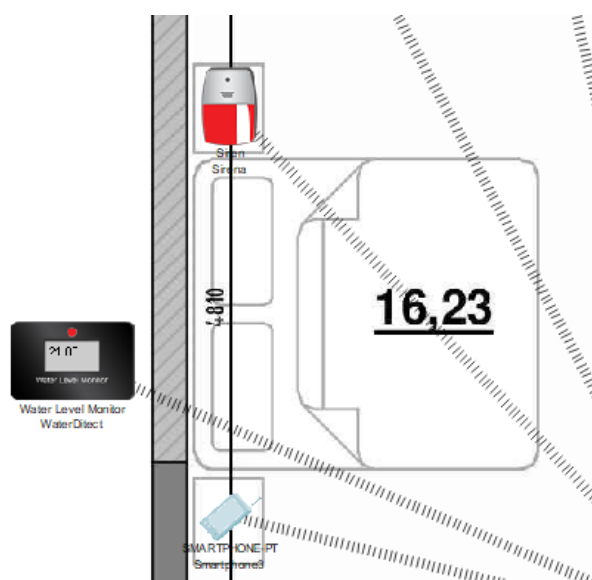


Рисунок 3.17 Тест 2

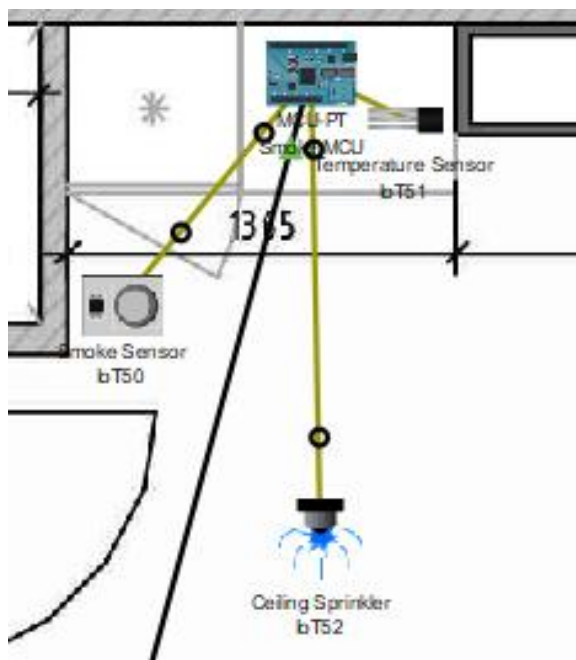


Рисунок 3.19 Тест 4

Висновки

В даному розділі було:

- Описано основні компоненти системи та причини їх вибору;
- Описано програмне забезпечення для реалізації проєкту та причини його використання;
- Проведено дослідження інформаційної системи та запрограмовано плати Arduino Uno на мові Java Script;
- Проведено тести для перевірки працездатності створеної інформаційної системи.

ВИСНОВОК

Головна мета даної роботи – розробка інформаційної системи за допомогою пристроїв IoT для сповіщення небезпечної ситуації. В ході роботи були сформовані головні вимоги до системи та шляхи їх реалізації та втілення в життя.

Порівнюючи існуючі на сьогодні інформаційної системи сповіщення безпеки можна зробити висновок, що найбільш перспективними є інтегровані системи керування пристроями IoT та використання для цього плат Arduino, які мають достатню кількість безсумнівних переваг:

1. Дана система не є новою, тому вже певний час вдало впроваджується, тому була проведена робота з коригування помилок та незручностей, які виникали при її використанні та інтеграція нових технічних засобів.

2. Створення єдиних загальноприйнятих стандартів, які були схвалені багатьма розробниками інтелектуальних систем.

3. Велика економічна вигода для споживачів та розробників.

Завдяки цьому саме ця система була вибрана як основа для власної роботи. Був створений макет з використанням певних технічних засобів. Після проведення тестувань можна стверджувати, що система працює коректно.

По ходу використання роботи були отримані певні результати:

1. Після проведення аналізу сучасних інтелектуальних систем був обраний інструментальний засіб програмного забезпечення.

2. В ході даної роботи було запропоновано концепції для інформаційної системи захисту життя людини під час небезпечних ситуацій, а саме: пожежа, викид шкідливих газів, повінь, грабіж.

3. Було налаштовано IoT девайси та з'єднання з домашнім шлюзом. Запрограмовано плати Arduino для правильної роботи датчиків

4. Розроблена фізична моделі розумного дому. Як результат дана система допоже забезпечити людям миттєве інформування про безпеки

прив'язуючись до геолокації помешкання, що зменшить панічні настрої та зможе зберегти людське життя.

5. За результатами тестування можна зробити висновок, що розроблена система відповідає технічним вимогам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Е.А. Тесля. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Е.А. Тесля – Санкт Петербург, 2008.
2. Т. Р. Элсенпитер, Дж. Велт. «Умный Дом строим сами» / Т. Р. Элсенпитер, Дж Велт/ КУДИЦ-ОБРАЗ. 2005.
3. В.Н. Харке «Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве» / В.Н. Харке– М.: Техносфера, 2006.
4. Т. Р. Элсенпитер, Дж. Велт. «Умный Дом строим сами» / Т. Р.Элсенпитер, Дж Велт / КУДИЦ-ОБРАЗ. 2005.
5. В.Н. Гололобов. «Умный дом» своими руками. / В.Н. Гололобов – М.: НТ Пресс, 2007.
6. М. Э. Сопер. Практические советы и решения по созданию « Умного дома » / М. Э. Сопер. – М.: НТ Пресс, 2007.
7. Arduino Uno R3 Technical Details; веб-сайт. URL: <http://static6.arrow.com/aropdfconversion/97b354c3cd11851042ebd1fd252fbd6b6def5a73/adafruit2877arduinouno.pdf>.
8. Arduino Nano. User Manual веб-сайт. URL: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>