

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
(повне найменування факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра
(ступінь вищої освіти)

на тему «Розробка системи автоматизації рутинних офісних процесів за
допомогою іot-технологій»
(назва теми)

Виконав: студент 4 курсу, групи КНТ-519

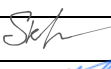
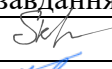
Спеціальності _____
123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерна інженерія

МІНЧЕНКО Є.О.
(ПРІЗВИЩЕ та ініціали)

Керівник	<u>СКРУПСЬКИЙ С.Ю.</u> (ПРІЗВИЩЕ та ініціали)
Рецензент	<u>КАЧАН О.І</u> (ПРІЗВИЩЕ та ініціали)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
	СКРУПСЬКИЙ С.Ю., к. т. н., доцент		
Нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання «8» травня 2023року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
	Аналіз технічного завдання	1 тиждень	виконано
	Аналіз структури роботи сигналізацій для автотранспорту	1 тиждень	виконано
	Вибір відповідної технології передачі інформації	1 тиждень	виконано
	Підбір комплектуючих елементів для створення системи, схеми підключення	2 тиждень	виконано
	Написання програмного коду	3 тиждень	виконано
	Проектування програмних засобів	3 тиждень	виконано
	Реалізація проєкту	3 тиждень	виконано
	Проходження нормоконтролю	4 тиждень	виконано
	Захист роботи	4 тиждень	виконано

Студент(ка)

 Євген МІНЧЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

 Степан СКРУПСЬКИЙ
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 60 с., 10 рис., 14 джерел.

ІОТ, ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ, ОФІСНЕ ПРИМІЩЕННЯ

Мета випускної роботи полягає у розробці програмного та апаратного комплексу на базі концепції IoT для автоматизації процесу зміни товарів споживання, зменшення тимчасових та фінансових витрат.

Завданнями випускної роботи є:

- проаналізувати існуючі методи з доставки та зміни товарів, запропонувати свій;
- вивчити історію становлення IoT-концепції;
- проаналізувати та вибрати hardware-пристрої;
- проаналізувати та вибрати інструментарій для створення програмного комплексу;
- розробити архітектуру та програмну частину hardware-пристрою;
- розробити програмну частину, використовуючи інструментарій Об'єктом дослідження є використання IoT-концепції для;
- автоматизації процесів у сфері життя офісних приміщень.

Об'єктом дослідження є використання IoT-концепції для автоматизації процесів у сфері життя офісних приміщень.

Предмет дослідження становлять програмний та апаратний комплекс для автоматизації зміни продуктів харчування.

Методи дослідження. Були використані поняття із середовища IoT, hardware-проекування, web-програмування, методи статистики та аналізу даних, літератури.

ЗМІСТ

Вступ.....	6
1 Проблема поповнення продуктових товарів в офісах та методи вирішення	8
1.1 Аналіз технічного завдання.....	8
1.2 Вимогу до продукту що розробляється.....	12
1.3 Пропонований метод.....	15
2 Огляд та вибір програмного та апаратного інструментарію.....	20
2.1 Пристрій Arduino	20
2.2 Пристрій Raspberry Pi	25
2.3 Вибір пристрою для реалізації проекту	29
2.4 Вибір датчиків для реалізації системи	32
2.5 Висновок до розділу.....	34
3 Програмний інструментарій.....	35
3.1 Веб-частина.....	35
3.2 Бот Microsoft bot framework	42
3.3 Хмарні платформи.....	43
3.4 Висновки до розділу	47
4 Розробка програмної та апаратної частини	47
4.1 Апаратна частина	47
4.2 Програмна частина.....	54
4.3 Висновки розділу.....	57
Висновки	58
Перелік джерел посилання	59
Перелік графічного матеріалу:	
Пл1 – Графік ринку IoT	
Пл2 – Технічні характеристики плат	
Пл3 – Схема зв'язку системи	
Пл4 – Хід розробки системи;	

ВСТУП

Ця робота присвячена дослідженню в галузі автоматизації процесів із застосуванням технології Інтернет-речей.

IoT – відносно молода концепція, сформульована Кевіном Ештоном у 1999 році, яка росте дуже швидко. Ще в 2010 році, за даними аналітичного відділу Cisco, кількість пристроїв, підключених мережі інтернет, удвічі перевищило населення людства[1]. У 2011 році компанія Gartner, що займається дослідженням та консалтингом ринку, помістила IoT у свій, так званий, «цикл зрілості технологій» на етап «технологічного тригера», який триває 10 років, що означає, технологія переросла забобони та «гаражний» рівень, інвестори і обивателі вірять у неї. [2] Juniper Research прогнозує зростання до 2021 року до 46 мільярдів штук та ринок у 1,7 трильйонів доларів [3]. Все сказане вище призводить до того, що великі компанії на ринку почали активно вкладати в даний напрямок, розробляти стандарти, впроваджувати на виробництвах.

Автоматизація процесів з'явилася досить давно і має прості цілі: підвищення різноманітних параметрів виробництва, скорочення чисельності персоналу, зменшення витрат. Разом з Інтернетом-речевою можна зосередитися на двох останніх. На жаль, більшість досліджень та робіт присвячені застосуванню напрацювань на промислових заводах та інших великих промислових підприємствах, але не варто залишати за бортом офісних робітників, там теж є де розвернутися.

В контексті концепцій автоматизації технологічних процесів і IoT було обрано нижній рівень автоматизації, на рівні виконавця, третьої групи, тобто для підприємств сфери послуг.

Проаналізувавши спектр існуючих проблем в офісних приміщеннях, автор вирішив зупинитися на одному з обов'язків, який виконує співробітник, який працює на позиції офіс-менеджера, тобто людини, яка стежить за належним функціонуванням офісу. У набір його функцій входять: постачання працівників

необхідними канцтоварами, зустріч відвідувачів, організація нарад, реєстрація кореспонденції, деяка секретарська робота, і, нарешті, забезпечення снеками колег чи гостей. Вибір впав на останній пункт, тому що це те, що закінчується найшвидше, присутні не лише витрати на сам товар, а й на доставку.

Актуальність випускної роботи характеризується бажанням людей мінімізувати витрати як фінансові, так і людські на рутинні процеси, особливо в середовищі It-компаній, де кожна хвилина висококваліфікованого фахівця на рахунок.

Мета випускної роботи полягає у розробці software та hardware комплексу на базі концепції IoT для автоматизації процесу зміни товарів споживання, зменшення тимчасових та фінансових витрат.

Завданнями випускної роботи є:

- проаналізувати існуючі методи з доставки та зміни товарів, запропонувати свій;
- вивчити історію становлення IoT-концепції;
- проаналізувати та вибрати hardware-пристрої;
- проаналізувати та вибрати інструментарій для створення програмного комплексу;
- розробити архітектуру та програмну частину hardware-пристрою;
- розробити програмну частину, використовуючи інструментарій Об'єктом дослідження є використання IoT-концепції для;
- автоматизації процесів у сфері життя офісних приміщень.

1 ПРОБЛЕМА ПОПОВНЕННЯ ПРОДУКТОВИХ ТОВАРІВ В ОФІСАХ ТА МЕТОДИ ВИРІШЕННЯ

1.1 Аналіз технічного завдання

Проблема поповнення продуктових товарів в офісах може виникати через брак або неправильний облік товарів, неефективні процеси замовлення та постачання, а також недостатню координацію між різними відділами. Однак існують різні методи вирішення цієї проблеми. Ось кілька підходів, які можуть допомогти в автоматизації та оптимізації процесу поповнення продуктових товарів в офісах. Використання системи обліку запасів: Встановіть систему обліку запасів, яка дає змогу відстежувати кількість і стан товарів. Такі системи можуть бути у вигляді програмного забезпечення або хмарних сервісів, які автоматично реєструють продажі, зчитують штрих-коди або використовують інші методи для відстеження рівня запасів. Це допоможе вам знати, коли і які товари потрібно замовляти.

Встановлення автоматичної системи замовлення: Розгляньте можливість встановлення автоматичної системи замовлення, яка буде автоматично запитувати поповнення запасів, коли вони досягають визначеного рівня. Це може бути пов'язано з вашою системою обліку запасів або працювати незалежно. Такі системи можуть бути інтегровані з постачальниками або використовувати онлайн-платформи для спрощення процесу замовлення.

Централізація та координація: Створіть централізовану систему управління поповненням запасів, яка об'єднує всі офіси та відділи. Це дасть змогу краще планувати та синхронізувати потреби в поповненні товарів між різними місцями. Централізована система може включати центральний склад або використовувати механізми комунікації для обміну інформацією про запаси та замовлення.

Автоматизація сповіщень: Налаштуйте автоматичні сповіщення та нагадування для контролю запасів. Це може бути у вигляді електронних

повідомлень, SMS-повідомлень або сповіщень у додатках. Такі повідомлення допоможуть вам бути в курсі поточного стану запасів і вживати своєчасних заходів.

Аналіз та оптимізація процесів: Проведіть аналіз поточних процесів поповнення товарів і визначте вузькі місця та неефективності. Використовуйте отримані дані для оптимізації процесів і поліпшення ефективності. Може бути корисно впровадити методи JIT (Just-In-Time), які дають змогу замовляти товари тільки в потрібній кількості і в потрібний час, мінімізуючи зберігання запасів.

Встановлення автоматичних систем доставки: Розгляньте можливість встановлення автоматичних систем доставки, таких як автоматичні диспенсери або торговельні автомати, які надають доступ до продуктів і контролюють їх витрачання. Це може допомогти скоротити час і зусилля, що витрачаються на поповнення товарів.

Успішне розв'язання проблеми поповнення продуктових товарів в офісах вимагає комплексного підходу, що поєднує автоматизацію, координацію та оптимізацію процесів. Використовуючи відповідні інструменти та методи, ви можете спростити процес поповнення товарів, скоротити тимчасові витрати і поліпшити ефективність у вашому офісі.

У зв'язку бурхливо зростаючим темпом збільшення іт-компаній, компаній, де людські кадри ставляться найвище, постає питання: як утримати співробітника крім високої зарплати, цікавих проектів? Логічна відповідь: зробити побут якнайкраще, щоб він міг почуватися як удома, звідси і випливає те, що підприємства створюють розважальні простори, стенди зі снеками та інше. Дедалі більше людей починають працювати у офісах, ніж заводах чи сільськогосподарських масивах[4]. За даними журналу the Village в офісі Google працюю 150 осіб, там розташовано все, що було сказано вище для зручності співробітників, так як дана компанія є маяком для інших, не важко уявити і логічно припустити, що інші, щоб утримати поточних співробітників і залучати собі нові кадри, будуть наслідувати приклад [5]. Але варто подивитися трохи в інший бік, існує практика гостям у компанії пропонувати чай, каву, солодоці, з

цього впливає, що не тільки необхідно доставляти, під час заповнювати снекі співробітникам, що позитивно впливатиме на їх працездатність[6], але і на також на імідж компанії. Все це призводить до функції людини, яка працює на позиції офіс-менеджера. У його обов'язки входить придбання за рахунок коштів компанії продуктів харчування даного типу, так як вигідно закуповувати у великих кількостях, щоб не втрачати час і гроші на доставку, основу снеків складають печива, солодоші, продукти, що довго зберігаються, що негативно впливає на здоров'я співробітників [7]. Під час заповнювати снеки співробітникам, що позитивно впливатиме з їхньої працездатність[6], а й у також імідж компанії. Все це призводить до функції людини, яка працює на позиції офіс-менеджера. У його обов'язки входить придбання за рахунок коштів компанії продуктів харчування даного типу, так як вигідно закуповувати у великих кількостях, щоб не втрачати час і гроші на доставку, основу снеків складають печива, солодоші, продукти, що довго зберігаються, що негативно впливає на здоров'я співробітників [7]. . під час заповнювати снеки співробітникам, що позитивно впливатиме з їхньої працездатність[6], а й у також імідж компанії. Все це призводить до функції людини, яка працює на позиції офіс-менеджера. У його обов'язки входить придбання за рахунок коштів компанії продуктів харчування даного типу, так як вигідно закуповувати у великих кількостях, щоб не втрачати час і гроші на доставку, основу снеків складають печива, солодоші, продукти, що довго зберігаються, що негативно впливає на здоров'я співробітників [7].

Все це призводить до того, що необхідно створити сервіс, який займатиметься доставкою до офісів здорових продуктів. Тим більше, що тренд на здорову їжу тільки починає набирати популярності, вже зараз, за даними Forbes, ринок становить 1 трильйон доларів [8].

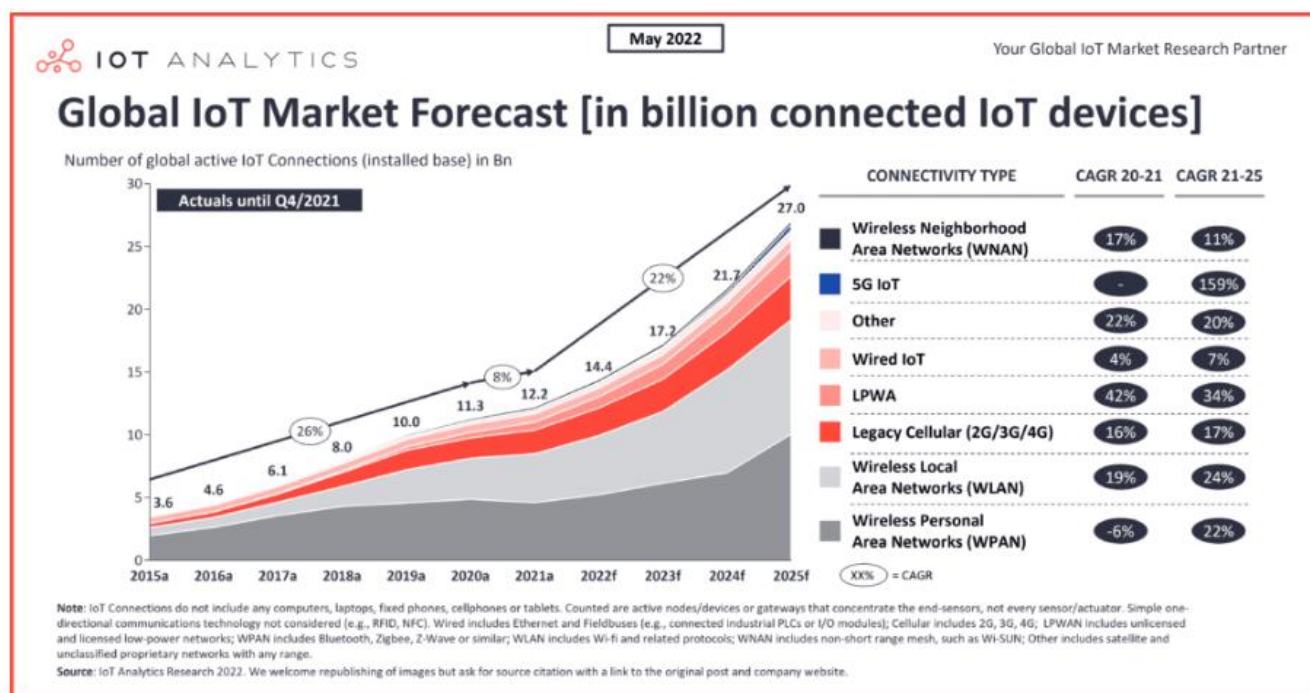


Рисунок 1.1 – Графік ринку IoT

Другим вагомим аргументом є швидкий і стрімкий перехід до 4 промислової революції — упровадження комп'ютеризації, бездротового інтернету, зміна ринкових відносин на модель послуг. У цьому числі сюди включається поняття Промисловість 4.0[9]. Джерело: <http://genesisnanotech.com/tag/fourth-industrial-revolution/>

| From Industry 1.0 to Industry 4.0

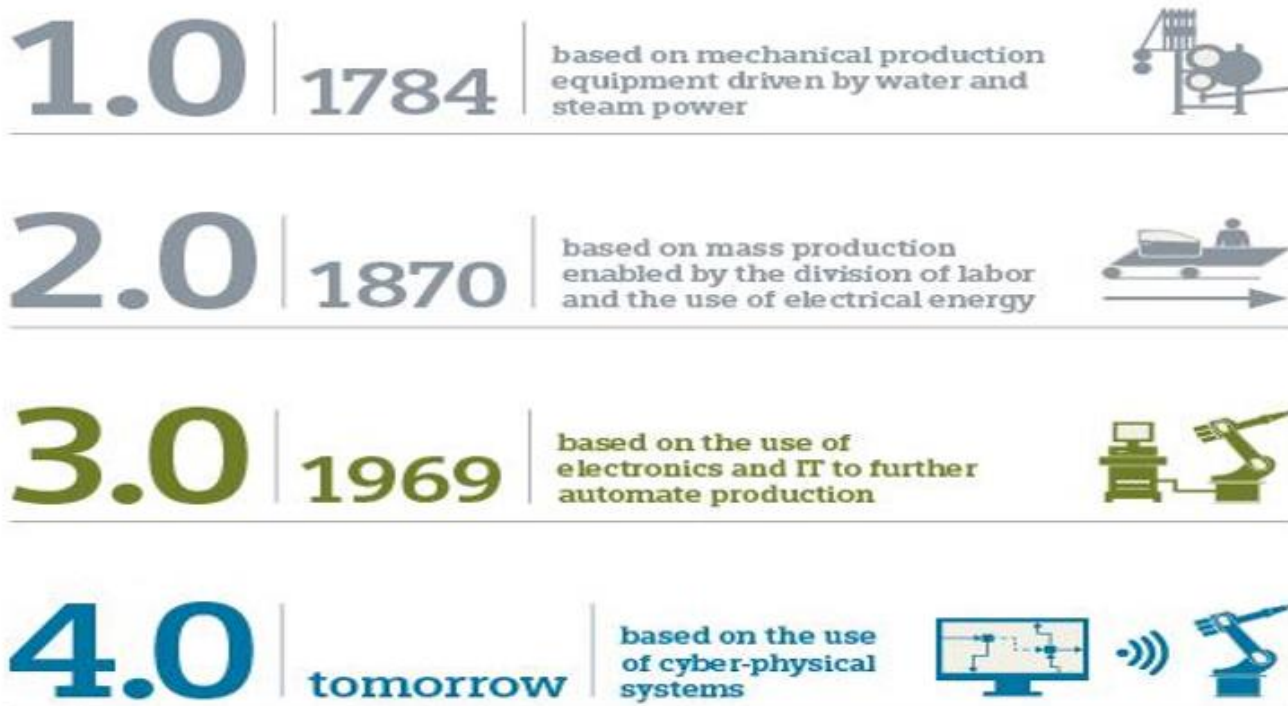


Рисунок 1.2 - Індустрія 4.0

1.2 Вимогу до продукту що розробляється

Один з найпростіших способів, що напрошується, але в той же час має безліч мінусів. Дивлячись з позитивного боку, можна сказати, що співробітник може приблизно розрахувати скільки потрібно продуктів купити. З мінусів: людина змушена вічно замислюватися над асортиментом, витрачаючи на цей робочий час, якщо купувати вічно одне й теж, може, ніхто і не буде скаржитися, але атмосфера в колективі з часом трохи погіршиться, потрібно багато місця, щоб зберігати продукти, інакше доведеться витрачати час та гроші на те, щоб повторити операцію. На жаль, немає жодних даних, скільки ж насправді витрачають компанії на подібний зміст співробітників, тому дослідження було проведено емпіричним шляхом. Вихідні дані: 2 особи, офіс у центрі міста. Було обрано оптовий магазин і закуплено товар, асортимент різний, вага 5 кг, в

середньому за перекушування з'їдається 70 грам продукту людиною, так як це все швидкі вуглеводи, ними складно насититися, таких перекусів за день 3 штуки. Подібна кількість товару було витрачено приблизно за місяць (27 робочих днів).

Цей тип насамперед відрізняється тим, що компанія не платить ні за установку, ні за снеки, все фінансове навантаження лягає на плечі співробітників. Відповідно всі плюси і створення сприятливої атмосфери випаровуються. Мабуть, єдиний позитивний момент- відсутність участі офіс-менеджера, якась автоматизація з отримання товару, знову ж таки, заповнює автомат співробітник вендінгової компанії та за певними числами.

Якщо співробітникам сподобався один вид батончиків і він швидко закінчився, компанія постачальник нічого про це не дізнається, і поповнить апарат лише наступного місяця. Відбувається простий.

Найчастіше компанії укладають договір з так званою доставкою в офіс, існують різні варіанти як традиційні з печивами, так і більш трендові зі здоровою їжею. Усі мінуси печива та подібних солодошів були розглянуті у параграфі 1.2 про можливість закупівлі офіс-менеджером, тому перейду до другої, а саме сервісів з доставки здорової їжі. За твердженням компанії Brainfoods, їх бокс з 90 снеків кожен по 30 грам, що в сумі дає 2.7 кілограм вистачає на місяць їдальні або 2 тижні в багато відвідуваних закладах, якщо точніше, то 6 осіб можуть харчуватися 3-4 тижні, доставка безкоштовна.



Рисунок 1.3 – Пропозиція cheku

Другий представник цієї галузі компанія Snackis, тут стенд для снєків в кількості 400 штук для людей від 1 до 15 коштує близько 5000 гривень, доставка здійснюється щомісяця.



Рисунок 1.4 – Пропозиція Snackis

Розглянемо рішення. Плюси полягають у тренді на здоровий спосіб життя, турботі про співробітників, непоганому дизайні. Мінуси: доставка щомісяця, товари можуть бути з'їдені раніше, або залежатися, досить висока вартість, немає об'єктивної розграмовки.

1.3 Пропонований метод

Вплив IoT (Інтернету речей) на автоматизацію рутинних процесів.

Останніми роками IoT (Інтернет речей) став одним із найбільш обговорюваних і перспективних технологічних напрямів. IoT являє собою мережу фізичних пристроїв, з'єднаних один з одним і з інтернетом, які здатні збирати і

обмінюватися даними. Ця технологія має величезний потенціал у багатьох сферах, включно з автоматизацією рутинних процесів. У цій статті ми розглянемо вплив IoT на автоматизацію рутинних процесів і переваги, які вона може принести.

Розумний будинок. Однією зі сфер, де IoT має найбільший вплив, є розумний дім. За допомогою IoT-пристроїв, будинок може бути повністю автоматизований і керований через інтернет. Пристрої, такі як розумні термостати, освітлення, замки, системи безпеки і домашні асистенти, можуть бути пов'язані в одну мережу і управлятися віддалено. Наприклад, ви можете налаштувати систему автоматичного керування освітленням, щоб вмикати світло, коли ви приходите додому, або встановити термостат, який самостійно регулює температуру в приміщенні залежно від наявності людей.

Розумний офіс. Вплив IoT на автоматизацію рутинних процесів поширюється і на робоче середовище. Розумні офіси використовують IoT-пристрої для підвищення ефективності та комфорту працівників. Наприклад, пристрої для управління освітленням і кліматом можуть бути налаштовані на автоматичну оптимізацію енергоспоживання відповідно до наявності співробітників. Системи розумних будівель також дають змогу автоматично відстежувати й контролювати використання ресурсів, як-от вода й електрика, що сприяє економії та зниженню негативного впливу на навколишнє середовище.

Промислова автоматика. IoT має велике значення в галузі промислової автоматизації. Безліч сенсорів і пристроїв IoT можуть бути встановлені на обладнання та машинне устаткування, щоб відстежувати і контролювати його стан. Це дає змогу запобігати аваріям і збоям, а також оптимізувати процеси виробництва. Наприклад, IoT-сенсори можуть контролювати рівень запасів на виробництві та автоматично запитувати поповнення, коли запаси вичерпуються. Це дає змогу знизити витрати на ручну працю і скоротити час простою обладнання.

Транспортні системи. IoT також має значний вплив на автоматизацію рутинних процесів у транспортних системах. Розумні транспортні системи, що використовують IoT, дають змогу відстежувати і керувати автопарками,

маршрутами і дорожнім рухом. Транспортні засоби, оснащені сенсорами і пов'язані з інтернетом, можуть попереджати про несправності, оптимізувати маршрути і покращувати безпеку на дорогах. Крім того, IoT дає змогу автоматичне збирання та аналіз даних про транспортні потоки, що допомагає в управлінні міським рухом і розробці більш ефективних транспортних рішень.

Переваги IoT для автоматизації рутинних процесів. Ефективність: IoT-технології дають змогу автоматизувати процеси, що покращує їхню ефективність і знижує ймовірність помилок, пов'язаних із втручанням людини.

Скорочення витрат: Автоматизація за допомогою IoT дає змогу скоротити витрати на ручну працю та управління, а також оптимізувати використання ресурсів.

Покращена безпека: IoT-пристрої можуть попереджати про можливі проблеми та аварії, що сприяє підвищенню безпеки на робочих місцях і в інших сферах.

Комфорт і зручність: IoT-технології дають змогу створювати комфортніші та зручніші умови для життя і роботи, керувати пристроями віддалено і передбачати потреби користувачів.

Інтелектуальний аналіз даних: Зібрані пристроями IoT дані можуть бути проаналізовані для виявлення тенденцій, патернів і прогнозування потреб, що допомагає в ухваленні більш поінформованих рішень.

Екологічна відповідальність: IoT дає змогу контролювати й керувати споживанням ресурсів, як-от електрика і вода, що сприяє зниженню негативного впливу на навколишнє середовище. Розширення можливостей: IoT створює нові можливості для розвитку інновацій та створення нових продуктів і послуг, заснованих на автоматизації та мережевій взаємодії.

Вплив IoT на автоматизацію рутинних процесів неможливо переоцінити. Ця технологія відкриває нові горизонти в галузі автоматизації, підвищуючи ефективність, скорочуючи витрати і створюючи комфортні умови для життя і роботи. Однак, впровадження IoT вимагає відповідної інфраструктури, мережевого забезпечення та безпеки для захисту даних і запобігання

вразливостей. З розвитком IoT очікується подальше поліпшення автоматизації рутинних процесів і створення більш інтелектуальної та пов'язаної мережі пристроїв, яка буде продовжувати перетворювати наше повсякденне життя і робоче середовище.

Вперше термін був використаний учасником research-групи Auto-ID при MIT Кевіном Ештоном під час доповіді щодо впровадження радіочастотних міток.

American опублікував статтю, так її і назвавши, в якій йдеться можливостях цієї концепції, наведено приклади того, як всілякі побутові предмети взаємодіють між собою. Автоматизація подібних процесів - ідея не найновіша, але об'єднавши всі предмети в одну мережу інтернет-протоколом, можна створити щось раніше небачене так і сталося. Знаковим періодом вважається 2008-2009 роки, коли кількість підключених до мережі інтернет пристроїв перевищила кількість жителів на планеті Земля, так вважають аналітики компанії Cisco. Але головний виток у розвитку стався вже 2010 року, тоді IoT став флагманом "туманних обчислень", що дозволяє перевести всі обчислювальні потужності із "землі".

IoT -Є мережа фізичних предметів, об'єднаних між собою інтернет-протоколом, що усуває з ланцюжка дій людський фактор, переважно передбачається, що всі обчислення відбуваються в «хмарі».

Smart-city –своєрідне уявлення про містобудування, коли комунікації, інформаційні технології та IoT поєднані для вирішення міських проблем та управлінням життєзабезпеченням городян. Під Казанням будується свій Smart-City.

Internet of Humans –концепція, яка прийде на зміну IoT, яка говорить про те, що нам вже не потрібні будуть всілякі гаджети, що переносяться, ми самі станемо ними, за допомогою вживлення різних датчиків.

Пропонується взяти всі плюси існуючих методів та об'єднати їх. Так має бути створений сервіс, до якого включається апаратний та програмний комплекс. До складу входить, так званий, SmartContainer, що є диспенсером для пластівців, горіхів, датчиками відстеження стану наповненості, веб-частина, на якій користувач може відслідковувати стан контейнера, бот для швидкого доступу та оповіщення про зміну.

З боку користувача: Користувач вибирає на сайті набір (є кілька варіантів) та кількість місяців для підписки (1, 3, 6, 12). Апарат встановлюється в офісі, підключається до мережі. Контейнери заповнюються. У момент, коли ступінь наповненості досягає критичної позначки (10%), клієнту в телеграм надходить повідомлення про те, що контейнери будуть змінені протягом якогось часу. Після цього приїжджає людина, змінює контейнери (наповнювач вже всередині). Робота продовжується.

З технічної сторони: після встановлення апарату та вибору користувачем тарифу та набору. При включенні пристрою запускається GSM-модуль, раз на 5 хвилин зчитуються дані (відстань від датчика до мети, наповнювача) за допомогою датчиків оптичної порожнечі, раз на 60 хвилин дані відправляються на сервер-балансир, який існує для фільтрації запитів і розмежування прямого доступу до пристрої мережі Інтернет, попередньо GSM-модуль перемикається в режим передачі даних. На стороні сервера перевіряється ступінь наповненості контейнерів, якщо один з них досяг критичної позначки, відправляється сигнал співробітнику про те, що потрібно змінити, також телеграм відсилається користувачеві, що контейнер буде поміняний протягом якогось часу.

Було проведено досвід, обсяг контейнерів налічує 6 літрів, що дорівнює 6 кілограма ваги продукту, у магазині за комерційними цінами було закуплено товар. До складу входять різні горіхи та сухофрукти. На відміну від печива та інших солодощів, ця їжа засвоюється досить довго, що дозволяє більше часу почуватися ситим, так само має велику калорійність, скорочуючи споживання.

Так для двох осіб 6 кг вистачило приблизно 2 місяці, з умовою споживання 100 грамів раз на два рази на день.

Існуючі способи або дорогі, або локальні, або застаріли. Пропонований поєднує в собі потяг людства до здорового харчування, нові технології, і що найважливіше для бізнесу: скорочення фінансових витрат, так з усіх перерахованих вище варіантів, запропонований став найдешевшим. Так подібний пристрій і разом із сервісом вирішує проблему доставки та своєчасну зміну продукту, не даючи йому втратити свої смакові та товарні якості.

2 ОГЛЯД ТА ВИБІР ПРОГРАМНОГО ТА АПАРАТНОГО ІНСТРУМЕНТАРІЮ

2.1 Пристрій Arduino

Використання Arduino для Інтернету речей (IoT).

Інтернет речей (IoT) - це мережа фізичних пристроїв, під'єднаних до інтернету і здатних обмінюватися даними. Arduino - це платформа для створення прототипів і розробки електронних проєктів. Поєднання цих двох технологій відкриває нові можливості для створення пристроїв і систем, пов'язаних з IoT. У цій статті ми розглянемо використання Arduino для IoT і переваги, які воно може надати.

Простота використання. Arduino надає просте і зрозуміле середовище розробки, що робить його ідеальним інструментом для початківців і досвідчених розробників. Він оснащений простим у використанні мікроконтролером і багатим вибором сенсорів, модулів і компонентів. Arduino також підтримує різні мови програмування, включно з Arduino IDE і мовою Wiring, що дає змогу розробникам легко створювати програми для керування пристроями IoT.

Гнучкість і масштабованість. Arduino пропонує гнучкість у виборі компонентів і модулів для створення пристроїв IoT. З його допомогою можна створювати прості прототипи і розширювати функціональність у міру необхідності. Arduino також може бути інтегрований з іншими платформами та сервісами, як-от Raspberry Pi, хмарні платформи та веб-сервіси, що дає змогу створювати складні та масштабовані системи IoT.

Взаємодія з навколишнім середовищем. Arduino підтримує безліч сенсорів і модулів, які дають змогу зчитувати дані з навколишнього середовища. Наприклад, за допомогою датчиків температури, вологості, освітленості та руху, Arduino може отримувати інформацію про стан навколишнього середовища. Ці дані можуть бути використані для автоматизації рутинних процесів, моніторингу та управління різними пристроями.

Віддалене керування та моніторинг. З використанням Arduino та інтернет-протоколів, як-от Wi-Fi або Ethernet, можна створити пристрої, якими можна віддалено керувати та моніторити через інтернет. Наприклад, за допомогою Arduino і відповідних модулів можна створити розумний дім, у якому можна керувати освітленням, термостатом, системою безпеки та іншими пристроями з будь-якого місця, під'єданого до інтернету.

Інтеграція з хмарними платформами. Arduino може бути інтегрований з різними хмарними платформами, такими як Amazon Web Services (AWS), Microsoft Azure, Google Cloud та іншими. Це дає змогу надсилати дані з пристроїв IoT у хмару для їхнього зберігання, аналізу та обробки. Хмарні платформи також надають можливості масштабування, безпеки та доступу до даних з будь-якої точки світу.

Застосування в різних галузях. Використання Arduino для IoT має широкі застосування в різних галузях. Наприклад, у сільському господарстві Arduino може бути використаний для моніторингу ґрунту, вологості, освітленості та автоматичного поливу. У медицині Arduino може використовуватися для створення пристроїв моніторингу здоров'я пацієнтів і передавання даних лікарям. У виробничій галузі Arduino може бути застосований для моніторингу та управління обладнанням, контролю якості та оптимізації виробничих процесів.

Використання Arduino для IoT відкриває нові можливості для створення інтелектуальних пристроїв і систем, які можуть поліпшити наше повсякденне життя і роботу. Його простота використання, гнучкість і можливість взаємодії з навколишнім середовищем роблять його привабливим вибором для розробки пристроїв IoT. Зі зростанням популярності IoT очікується подальший розвиток та інновації в галузі використання Arduino для створення розумних і пов'язаних пристроїв.

Arduino є популярною платформою для розробки IoT-проектів¹. Останні дослідження використання Arduino для IoT зосереджуються на розширенні можливостей платформи та полегшенні розробки проектів.

Наприклад, Arduino IoT Cloud є онлайн-платформою, яка дозволяє

створювати, розгортати та моніторити IoT-проекти. Ця платформа має зручний інтерфейс та є все-в-одному рішенні для конфігурації, написання коду, завантаження та візуалізації.

Іншим напрямком досліджень є розширення можливостей платформи за допомогою додаткових модулів¹. Наприклад, модуль LoRaWAN дозволяє забезпечити зв'язок між пристроями на великій відстані.

Також було проведено дослідження використання Arduino для створення систем автоматизації та контролю. У цьому дослідженні було показано, що Arduino може бути використана для створення систем автоматизації та контролю на основі зчитування даних з датчиків.

Arduino - це відкрита платформа для програмування та створення простих електронних пристроїв. Вона пропонує простий і доступний спосіб автоматизувати рутинні процеси та створювати інтерактивні системи.

Arduino складається з двох основних компонентів: мікроконтролера та середовища розробки. Мікроконтролер Arduino - це маленький комп'ютер, який можна програмувати для виконання різних завдань. Він має вбудовані входи і виходи, які дають змогу керувати різними електронними компонентами, такими як світлодіоди, датчики, мотори тощо.

Середовище розробки Arduino - це програмне забезпечення, яке дає змогу писати та завантажувати програми на мікроконтролер. Воно пропонує просту у використанні мову програмування і набір бібліотек, які спрощують взаємодію з різними компонентами і сенсорами.

За допомогою Arduino ви можете автоматизувати рутинні процеси в будинку, офісі або в інших сферах. Ось кілька прикладів використання Arduino для автоматизації:

Розумний дім: Ви можете використовувати Arduino для керування освітленням, термостатами, системами безпеки та іншими пристроями у вашому домі. Наприклад, ви можете налаштувати автоматичне увімкнення світла під час входу в кімнату або регулювання температури залежно від часу доби.

Автоматизація офісу: Arduino може допомогти вам керувати офісними

пристроями, як-от системами освітлення, кондиціонування повітря або розумними замками для безпеки. Ви можете створити розклад роботи пристроїв або налаштувати їх на реагування на певні умови.

Садівництво та зрошення: Arduino можна використовувати для автоматизації систем поливу, контролю вологості ґрунту та температури в теплицях. Ви можете налаштувати систему поливу для подачі води в певний час або залежно від показань датчиків.

Моніторинг і збір даних: Arduino дає змогу створювати системи моніторингу та збору даних. Ви можете використовувати різні датчики для вимірювання температури, вологості, освітленості та інших параметрів, а потім зберігати й аналізувати ці дані на комп'ютері або в хмарі.

Arduino - це гнучка та доступна платформа для автоматизації рутинних процесів. Вона дає змогу вам створювати власні пристрої та адаптувати їх під ваші потреби. З великим співтовариством розробників і ресурсами він також пропонує підтримку і можливості для розширення своїх проєктів.

Arduino –це друкована плата з власним процесором та пам'яттю (мікроконтролер). Також на ній є десяток контактів, до яких можна підключати всілякі датчики, сенсори. Усе це управляють у вигляді алгоритму, написаного розробником. Існує спеціальна IDE, яка постачається у комплекті з апаратною частиною. Хороший тим, що можна безперешкодно додавати нові компоненти, розширюючи пристрій. Виробляється кілька варіантів: Uno - стандартний тип, підходить майже для всього, Leonardo -Аналог Arduino Uno з змінним USB-UART інтерфейсом, Yun - Платформа, що об'єднує Arduino і необмежені можливості Linux для роботи з інтернетом , Mega - Розширена версія Arduino зі значною кількістю контактів, пам'яті та serial-портів, mini - Компактна модифікація Arduino Uno без власного USB-порту, micro - Еквівалент Arduino Leonardo у компактному корпусі та інші. Мікроконтролери для Arduino відрізняються наявністю попередньо прошитого у них завантажувача. За допомогою цього завантажувача користувач завантажує свою програму мікроконтролер без використання традиційних окремих апаратних програматорів.

Завантажувач з'єднується з комп'ютером через інтерфейс USB (якщо він є на платі) або окремим перехідником UART-USB. За замовчуванням мовою програмування є C++ з деякими особливостями, що допомагають у розробці. Але також існує безліч прошивок, які вирішують це питання.

Одним з найкращих і базових варіантів є вищезгадана Arduino Uno, приваблива своєю ціною, співвідношенням компактності кількості виходів/входів, відсутністю зайвих датчиків типу wi-fi. Arduino Uno побудована на базі мікроконтролера Atmega328.

Платформа має 14 цифрових введення/виходів (6 з яких можуть використовуватися як виходи ШІМ), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, силовий роз'єм, роз'єм ICSP та кнопку перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC, або акумуляторною батареєю.

Специфікація:

- мікроконтролер ATmega328;
- напруга живлення 5В;
- вхідна напруга (рекомендована) 7-12В;
- вхідна напруга (гранична) 6-20В;
- цифрове введення-виведення 14 лінії (6 з них = ШІМ);
- пналогове введення 6 ліній;
- постійний струм на лініях введення-виводу 40мА;
- постійний струм на лінії 3.3В 50мА;
- flash-пам'ять 32кб, 0.5 кб їх використані для завантажувача;
- SRAM-пам'ять 2кб;
- EEPROM-пам'ять 1кб;
- тактова частота 16МГц;

2.2 Пристрій Raspberry Pi

Використання Raspberry Pi для вирішення завдань у системі Інтернету речей.

Raspberry Pi - це одноплатний комп'ютер, розроблений для освітніх і розважальних цілей. Однак, завдяки своїй низькій вартості, гнучкості та потужності, Raspberry Pi став популярним вибором для розв'язання задач у системі Інтернету речей (IoT). У цій статті ми розглянемо використання Raspberry Pi для IoT і переваги, які він може надати.

Потужність і гнучкість. Raspberry Pi має достатню обчислювальну потужність і гнучкість для оброблення та передавання даних у системі IoT. Він оснащений потужним процесором, оперативною пам'яттю і можливістю підключення різних периферійних пристроїв. Завдяки цьому Raspberry Pi може виконувати складні обчислення, керувати сенсорами і модулями, а також підключатися до мережі для обміну даними.

Низька вартість. Однією з головних переваг Raspberry Pi є його низька вартість порівняно з традиційними комп'ютерами та серверами. Це робить його доступним для широкого кола розробників і людей, які цікавляться IoT. Ба більше, завдяки модульній архітектурі Raspberry Pi можна легко оновлювати та розширювати функціональність пристрою, не замінюючи всю систему.

Взаємодія з різними пристроями. Raspberry Pi може бути під'єднаний до різних пристроїв і сенсорів, що дає змогу зчитувати дані з навколишнього середовища та керувати зовнішніми пристроями. Наприклад, за допомогою Raspberry Pi можна створити систему розумного будинку, в якій можна керувати освітленням, термостатом, системою безпеки та іншими пристроями. Також Raspberry Pi може використовуватися в сільському господарстві, медицині, промисловості та інших галузях для збору та обробки даних.

Робота з хмарними платформами. Raspberry Pi може бути інтегрований з різними хмарними платформами, такими як Amazon Web Services (AWS),

Microsoft Azure, Google Cloud та іншими. Це дає змогу надсилати дані з пристроїв IoT у хмару для їхнього зберігання, аналізу та обробки. Хмарні платформи надають можливості масштабування, безпеки та доступу до даних з будь-якої точки світу. Raspberry Pi працює як міст між фізичними пристроями IoT і хмарною інфраструктурою.

Створення прототипів і навчання. Raspberry Pi ідеально підходить для створення прототипів IoT-пристроїв і для навчання в галузі IoT. Завдяки його доступності та гнучкості, розробники і студенти можуть експериментувати, створювати і тестувати свої ідеї, не витрачаючи великі суми грошей на обладнання та інфраструктуру. Raspberry Pi також має активну спільноту розробників і безліч ресурсів, що робить його чудовим вибором для навчання і самостійного дослідження в галузі IoT.

Використання Raspberry Pi для вирішення завдань у системі Інтернету речей надає безліч можливостей. Його потужність, гнучкість, низька вартість і здатність взаємодіяти з різними пристроями роблять його ідеальним інструментом для створення розумних пристроїв, моніторингу та управління процесами в різних галузях. З розвитком IoT очікується подальше розширення функціональності Raspberry Pi і його застосування в нових галузях.

Raspberry – на відміну від Arduino, носить горде ім'я мікрокомп'ютера, тобто має на собі вже повноцінну операційну систему, має більш високі технічні характеристики. Всупереч очікуванням автором, Pi створювався як навчальний інструмент для уроків інформатики, отримав колосальний успіх у співтоваристві, всього за 3 роки було продано понад 4.5 мільйонів пристроїв.

Raspberry Pi - це одноплатний комп'ютер, розроблений для навчання програмуванню та створення електронних проєктів. Він пропонує потужні обчислювальні можливості, безліч входів-виходів і підтримку різних операційних систем, що робить його ідеальним інструментом для автоматизації рутинних процесів.

Ось деякі особливості Raspberry Pi, які роблять його корисним для автоматизації. Обчислювальні можливості: Raspberry Pi оснащений процесором

ARM, який має достатню потужність для виконання різних завдань автоматизації. Він може запускати програми та скрипти, обробляти дані та взаємодіяти з іншими пристроями.

GPIO (General Purpose Input/Output) порти: Raspberry Pi має GPIO порти, які дають змогу під'єднувати різні електронні компоненти та пристрої. Ви можете використовувати ці порти для керування світлодіодами, реле, датчиками, моторами та іншими пристроями, необхідними для автоматизації.

Різноманітність операційних систем: Raspberry Pi підтримує різні операційні системи, включно з Raspbian (заснована на Linux), Ubuntu, Windows 10 IoT Core та інші. Це дає вам свободу вибору відповідної ОС для вашого проекту автоматизації.

Розробницькі можливості: Raspberry Pi має велику спільноту розробників, які активно створюють і діляться проектами та бібліотеками. Ви можете знайти готові рішення для автоматизації або створити власні, ґрунтуючись на доступних ресурсах і підтримці спільноти.

З Raspberry Pi ви можете автоматизувати різні рутинні процеси, включно з розумним будинком, системами моніторингу, автоматизацією офісу та багато іншого. Наприклад, ви можете налаштувати систему розумного будинку для керування освітленням, опаленням, безпекою та мультимедіа. Ви також можете створити систему моніторингу за допомогою камер і датчиків для контролю температури, вологості та інших параметрів.

Raspberry Pi пропонує гнучкість і можливості для створення різних проєктів автоматизації залежно від ваших потреб і навичок програмування.

Крім того, Raspberry Pi підтримує різні мови програмування, такі як Python, C/C++, Java та інші, що робить його доступним для широкого кола розробників з різним рівнем досвіду. Ви можете використовувати ці мови для написання програм, контролю зовнішніх пристроїв, зчитування даних із сенсорів і багато чого іншого.

Деякі додаткові переваги Raspberry Pi в контексті автоматизації рутинних процесів. Мережеві можливості: Raspberry Pi має вбудований Ethernet-порт або

може бути під'єднаний до бездротової мережі Wi-Fi, що дає йому змогу взаємодіяти з іншими пристроями та обмінюватися даними мережею. Це корисно для віддаленого управління і моніторингу автоматизованими процесами.

Розширюваність: Raspberry Pi підтримує різні додаткові модулі та шилди, які дають змогу розширити його функціональність. Наприклад, ви можете додати камеру для відеоспостереження або модуль Bluetooth для управління бездротовими пристроями.

Низька вартість: Raspberry Pi доступний за відносно низькою ціною, що робить його більш доступним для широкого кола користувачів. Це чудова альтернатива більш дорогим комп'ютерним системам під час розроблення автоматизованих рішень.

Загалом, Raspberry Pi надає потужні інструменти та ресурси для автоматизації рутинних процесів. Він дає змогу вам створювати розумні системи, збирати й аналізувати дані, керувати пристроями та багато іншого. Завдяки своїй гнучкості та доступності, Raspberry Pi стає популярним вибором для створення автоматизованих систем у різних галузях, від домашньої автоматизації до промислової автоматики.

Налічує у своєму арсеналі безліч моделей: A, A+, B, B+, 2B, Zero, 3B, Zero W, що відрізняються між собою технічними характеристиками. Випущено кілька поколінь Raspberry-Pis. Перше покоління (Raspberry Pi 1 Model B) було випущено у лютому 2012 року. За ним була проста і недорога модель A. У 2014 році було представлено на огляд пристрій з покращеним дизайном в Raspberry Pi 1 Model B+. Ці плати мають розміри приблизно з кредитною карткою і є стандартним форм-фактором. Через рік були випущені покращені моделі A + і B +. У квітні 2014 року для вбудованих додатків було випущено «обчислювальний модуль», а в листопаді 2015 року для США було випущено Raspberry Pi Zero з меншими розмірами та обмеженими можливостями вводу/виводу (I/O) та загального призначення вводу/виводу (GPIO). \$ 5. Raspberry Pi 2, до якого додали більше оперативної пам'яті, був проведений у лютому 2015 року. Raspberry Pi 3 Model B, представлений у лютому 2016 року, комплектується вбудованими WiFi, Bluetooth

та USB. Станом на січень 2017 року Raspberry Pi 3 Model B є найновішою моделлю Raspberry Pi. Ціна на Pi варіюється від 5 до 35 доларів США. Станом на 28 лютого 2017 року був запущений Raspberry Pi Zero W, ідентичний Raspberry Pi Zero, але має функції Wi-Fi та Bluetooth Raspberry Pi 3 за 10 доларів США.

Всі моделі оснащені системою Broadcom на чіпі (SoC), яка включає ARM-сумісний центральний процесор (CPU) і вбудований графічний процесор (GPU, VideoCore IV). Частота процесора варіюється від 700 МГц до 1,2 ГГц для Pi 3 та пам'яті на борту від 256 МБ до 1 ГБ ОЗУ. Карти Secure Digital (SD) використовуються для зберігання операційної системи та програмної пам'яті у форматах SDHC або MicroSDHC. Більшість плат мають від одного до чотирьох роз'ємів USB, HDMI та композитний відеовихід, а також 3,5-мм роз'єм для аудіо. Вихід нижчого рівня забезпечується кількома висновками GPIO, що підтримують загальні протоколи, такі як I²C. У B-моделей є 8P8C Ethernet-порт, а у Pi 3 та Pi Zero W є Wi-Fi 802.11n та Bluetooth. Фонд надає Raspbian, дистрибутив Linux для Debian, для завантаження, а також сторонні Ubuntu, Windows 10 IOT Core, RISC OS та спеціалізовані медіацентри. Він підтримує Python і Scratch як основну мову програмування з підтримкою багатьох інших мов.

2.3 Вибір пристрою для реалізації проекту

Arduino – це материнська плата мікроконтролера. Мікроконтролер - це простий комп'ютер, який може запускати одну програму за раз, знову та знову. Він дуже простий у використанні. Плата Arduino найкраще підходить для простих завдань, що повторюються: відкриття і закриття дверей гаража, читання температури зовнішнього повітря і звітність в Twitter. Плати Arduino є мікроконтролери, а не повні комп'ютери. Вони не запускають повну операційну систему, а просто виконують написаний код, оскільки їх прошивка інтерпретує його. Ви втрачаєте доступ до базових інструментів, що надаються операційною

системою, але, з іншого боку, простіше виконувати простий код простіше і це досягається без накладних витрат операційної системи. Головна мета плати Arduino - це взаємодія з датчиками та пристроями, тому вона чудово підходить для апаратних проєктів, у яких ви просто хочете, щоб речі реагували на різні показання датчиків та ручне введення. Це може здатися не таким великим, але насправді це дуже складна система, яка дозволяє краще керувати вашими пристроями. Це відмінно підходить для взаємодії з іншими пристроями та приводами, де повна операційна система буде надмірною для обробки найпростіших операцій читання та відповіді.

системою Linux, та можливість запускати кілька програм. Його важче використовувати, ніж Arduino. Для всіх цілей та завдань, Малина Пі є повністю функціональним комп'ютером. Він має всі атрибути комп'ютера, з виділеним процесором, пам'яттю та графічним драйвером для виведення через HDMI. Він навіть запускає спеціально розроблену версію ОС. Це спрощує встановлення більшості програм для Linux і дозволяє з легкістю використовувати Пі як працюючий медіа-стрімер або емулятор відеоігор.

Хоча Пі не пропонує внутрішнє сховище, ви можете використовувати картки SD як флеш-пам'ять для всієї системи, що дозволяє швидко замінювати різні версії операційної системи або оновлення програмного забезпечення для налагодження. Через незалежне підключення пристрою ви також можете налаштувати його для доступу через SSH або передати файли на нього за допомогою FTP. Raspberry Pi найкраще використовувати, коли вам потрібен повноцінний комп'ютер: водіння складнішого робота, виконання безлічі завдань, інтенсивні обчислення (як для біткойна або шифрування).

Живлення. Ці дві системи мають різні вимоги до харчування. Малина Пі вимагає постійної потужності 5, щоб залишатися включеною, і, крім того, її необхідно відключити за допомогою програмного процесу, такого як традиційний комп'ютер. З іншого боку, Arduino починає виконувати код при включенні і зупиняється, коли ви витягуєте вилку. Щоб додати функціональність, ви або підключаєтеся безпосередньо до контактів на платі Arduino, або називаєте

"екрани" на верхній частині базового блоку. Існують сотні екранів, кожен з яких призначений для виконання іншого завдання, сполучення з певними датчиками та роботи один з одним для створення повного блоку керування. Портативність – проблема з Pi, оскільки для цього потрібно більше, ніж просто підключення кількох акумуляторів типу AA. Пристрій вимагає, щоб ви встановили джерело живлення та деяке додаткове обладнання, щоб забезпечити його постійну споживану потужність. Процес на Arduino трохи простіше, тому що вам просто необхідний акумулятор, який підтримує напругу вище за певний рівень, а також базовий екран для управління живленням.

Навіть якщо живлення падає на Arduino, ви не отримаєте пошкоджену операційну систему та інші помилки програмного забезпечення. Він просто почне запускати код, коли знову підключиться.

Мережа. Pi має вбудований Ethernet-порт, який забезпечує легкий доступ до будь-якої мережі з невеликими настройками. Також неважко досягти бездротового Інтернету на Pi, вам просто потрібно купити USB Wi-Fi та встановити драйвер. Після підключення можна використовувати ОС для підключення до веб-серверів, обробки HTML-коду або публікації в Інтернеті. Ви можете навіть використовувати його як VPN або сервер друку.

На жаль, Arduino не побудований для підключення до мережі безпосередньо з коробки. Для налагодження правильного з'єднання потрібно трохи більше зусиль, хоча це можливо. Вам знадобиться додатковий чіп, оснащений Ethernet-портом, і вам потрібно буде запаяти та запрограмувати плату, щоб усе було запущено.

Датчики. Хоча у Pi та Arduino є кілька інтерфейсних портів, набагато простіше підключити аналогові датчики до Arduino. Мікроконтролер може легко інтерпретувати та реагувати на широкий діапазон даних датчиків, використовуючи код, який ви їм поклали, що робить його відмінним, якщо ви маєте намір повторити серію команд або відгукуватися на дані датчиків як засіб налаштування сервоприводів та пристроїв. З іншого боку, Pi вимагає, щоб програмне забезпечення ефективно взаємодіяло з такими пристроями, і це не завжди те, що

вам потрібно, якщо ви намагаєтеся поливати рослини.

Коли ви маєте вибрати Arduino? Коли основним завданням є зчитування даних датчика та зміна значень двигунів або інших пристроїв. Враховуючи низьке енергоспоживання Arduino та його підтримку, це також гарний вибір, якщо ваш пристрій постійно працюватиме і практично не потребує взаємодії. Коли ви повинні вибрати Raspberry? Якщо ви інакше виконували своє завдання за допомогою персонального комп'ютера. Pi спрощує управління безліччю операцій, незалежно від того, чи збираєтеся ви підключатися до Інтернету для читання запису даних, перегляду медіа будь-якого типу або підключення до зовнішнього дисплея.

Грунтуючись на всьому вищесказаному вибір робиться на користь Arduino, яка має більш гнучкі налаштування і суміщення сенсорів, меншими розмірами, економічним витратою енергії. У Raspberry-Pi є дуже багато непотрібних деталей, які тільки ускладняють розробку.

2.4 Вибір датчиків для реалізації системи

Мініатюрний модуль GSM/GPRS стільникового зв'язку на основі компонента SIM800L, розробленого компанією SIMCom Wireless Solutions. Стандартний інтерфейс керування компонентом SIM800L надає доступ до сервісів мереж GSM/GPRS 850/900/1800/1900МГц для надсилання дзвінків, SMS повідомлень та обміну цифровими даними GPRS. Поставляється з вбудованою антеною, також можна підключити додаткові антени для покращення якості сигналу. Для покращення якості сигналу приєднується антена.

Детальний опис:

Живлення модуля здійснюється за допомогою зовнішніх джерел або через DC-DC перетворювальні системи

Діапазон для живлення повинен бути в межах 3.7 - 4.2В. Варто відзначити,

що 5В від Arduino небезпечні для роботи модуля, оскільки приведуть його до поломки, а 3.3В допоможуть реагувати на команди користувача, але до мережі підключення буде відсутнє

Управління відбувається через UART, але для Rx і Tx використовується дільник напруги, щоб не проводилося навантаження на плату.

Низьке енергоспоживання модуля SIM800L дозволяє його успішно використовувати для техніки з акумуляторним живленням.

При подачі живлення модуль відразу починає шукати мережу. Коли її знайдено, починає подавати сигнал світлодіод, повільно блимаючи. Про відсутність мережі говорить цей світлодіод, але за допомогою швидкого миготіння лампочки DC/DC перетворювач перетворює вхідну напругу постійного струму у вихідну напругу постійного струму іншої величини. Такі перетворювачі можуть бути ізольованими (з гальванічною розв'язкою входу та виходу) та неізольованими (без гальванічної розв'язки). Спочатку перетворювач поставляється з об'єднаним загальним дротом.

Невеликі за розмірами електронні пристрої, здатні під впливом електромагнітного випромінювання у видимому, інфрачервоному та ультрафіолетовому діапазонах подавати одиничний або сукупність сигналів на вхід реєструючої або керуючої системи. Оптичні датчики реагують на непрозорі та напівпрозорі предмети, водяну пару, дим, аерозолі.

Оптичні датчики є різновидом безконтактних датчиків, оскільки механічний контакт між чутливою областю датчика (сенсором) та об'єктом, що впливає, відсутній. Ця властивість оптичних датчиків обумовлює їх широке застосування автоматизованих системах управління. Дальність дії оптичних датчиків набагато більша, ніж у інших типів безконтактних датчиків.

Характеристики

- діапазон вхідної напруги: 3,6-25 В;
- діапазон напруги: 3,3–25 В;
- мінімальна різниця напруги: 0,6 В;
- максимальний вихідний струм: 5 А;

- максимальна потужність: 25 Вт;
- частота перетворення: 350 кГц;
- габарити: 46×50×20 мм.

2.5 Висновок до розділу

	Arduino Uno	Raspberry Pi Model B
Price	\$30	\$35
Size	7.6 x 1.9 x 6.4 cm	8.6cm x 5.4cm x 1.7cm
Memory	0.002MB	512MB
Clock Speed	16 MHz	700 MHz
On Board Network	None	10/100 wired Ethernet RJ45
Multitasking	No	Yes
Input voltage	7 to 12 V	5 V
Flash	32KB	SD Card (2 to 16G)
USB	One, input only	Two, peripherals OK
Operating System	None	Linux distributions
Integrated Development Environment	Arduino	Scratch, IDLE, anything with Linux support

Рисунок 2.1 – Технічні характеристики плат

У цьому розділі було розглянуто два варіанти апаратної частини пристрою, проведено аналіз плюсів і мінусів, і обраний один з них. Також наведено короткий опис, використовуваних датчиків, аргументація та приклади, для яких з'являться в розділі номер 4.

3 ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ

3.1 Веб-частина

3.1.1 Ruby On Rails

Ruby - динамічна, рефлексивна, інтерпретована високорівнева мова програмування]. Мова володіє незалежною від операційної системи реалізацією багатопоточності, строгою динамічною типізацією, збирачем сміття та багатьма іншими можливостями. Мова слідує принципу «найменшої несподіванки»: програма повинна поводитися так, як очікує програміст. Однак у контексті Ruby це означає найменший подив не при знайомстві з мовою, а при його ґрунтовному вивченні. На противагу машинно-орієнтованим мовам, що працюють швидше, метою цієї розробки була мова, найбільш близька до людини. Будь-яка робота з комп'ютером виконується людьми і для людей, і необхідно піклуватися в першу чергу про зусилля людей, що витрачаються. Мова дозволяє максимально швидко і просто для людини виконати завдання, хоча, можливо, це потребує додаткового часу роботи комп'ютера. Має лаконічний та простий синтаксис, частково розроблений під впливом Ада, Eiffel та Python.

Дозволяє обробляти винятки у стилі Java та Python. Дозволяє перевизначати оператори, які насправді є методами.

Повністю об'єктно-орієнтована мова програмування. Всі дані Ruby є об'єктами в розумінні Smalltalk. Єдиний виняток – керуючі конструкції, які в Ruby, на відміну від Smalltalk, не є об'єктами. Наприклад, число "1" - це екземпляр класу Fixnum. Також підтримується додавання методів у клас і навіть у конкретний екземпляр під час виконання програми.

Не підтримує множинне успадкування, але замість нього може використовуватися концепція «домішок», що базується в даній мові на механізмі модулів.

Містить автоматичний збирач сміття. Він працює для всіх об'єктів Ruby, зокрема для зовнішніх бібліотек.

Створювати розширення для Ruby на Cі дуже просто частково через збирача сміття, частково через нескладний і зручний API.

Підтримує замикання із повною прив'язкою до змінних.

Підтримує блоки коду (код полягає в {...} або do...end).

Блоки можуть використовуватися в методах або перетворюватися на замикання. Цілі змінні у Ruby автоматично конвертуються між типами Fixnum (32-розрядні) і Bignum (більше 32 розрядів) залежно від їх значення, що дозволяє робити цілі математичні розрахунки з якоюсь великою точністю.

Не вимагає попереднього оголошення змінних, але для інтерпретатора бажано, щоб змінним надавалося порожнє значення nil (тоді інтерпретатор знає, що ідентифікатор позначає змінну, а чи не ім'я методу). Мова використовує прості угоди для позначення області видимості. Приклад: просто var – локальна змінна, @var – змінна екземпляра (член або поле об'єкта класу), @@var – змінна класу, \$var – глобальна змінна.

Ruby безпосередньо в мові реалізовано багато шаблонів проектування, наприклад, «одиначка» (singleton) то, можливо (хоч і зобов'язаний) реалізований додаванням необхідних методів одного конкретному об'єкту (див. нижче).

Перенесений на безліч платформ. Він розроблявся на Linux, але працює у багатьох версіях Unix, DOS, Microsoft Windows (зокрема, Win32), Mac OS, BeOS, OS/2 тощо.

Rails - фреймворк, написаний мовою програмування Ruby, реалізує архітектурний шаблон Model-View-Controller для веб-застосунків, а також забезпечує їх інтеграцію з веб-сервером і сервером баз даних. Основними компонентами додатків на Ruby on Rails є модель (англ. model), представлення (англ. view) та контролер (англ. controller). Ruby on Rails використовує REST-стиль побудови веб-додатків.

Модель надає іншим компонентам програми об'єктно-орієнтоване відображення даних (наприклад, каталог продуктів або список замовлень). Об'єкти моделі можуть здійснювати завантаження та збереження даних у реляційній базі даних, а також реалізують бізнес-логіку. Навколо Rails склалася

велика екосистема плагінів, які також називаються «джемами» (gem з англ. — «самоцвіт»). Для управління плагінами існує спеціальна система RubyGems.

3.1.2 Django

Django: Вивчення і застосування веб-фреймворку для розробки сучасних веб-додатків. Django - це популярний веб-фреймворк, розроблений на мові Python, який надає набір інструментів і функціональність для створення потужних і масштабованих веб-додатків. У цій статті ми розглянемо основи Django, його ключові особливості та переваги, а також розглянемо, як використання Django спрощує і прискорює процес розроблення веб-додатків.

Знайомство з Django. Django пропонує розробникам зручне та ефективне середовище для створення веб-додатків. Він заснований на концепції Model-View-Controller (MVC) і забезпечує чітку розділеність коду, що робить додатки більш підтримуваними і масштабованими. Django також підтримує використання шаблонів і URL-маршрутизації, що полегшує розробку та управління різними сторінками і функціональністю програми.

Автоматизована адміністративна панель. Однією з ключових переваг Django є вбудована адміністративна панель, що дає змогу розробникам легко створювати та керувати адміністративним інтерфейсом свого застосунку. Django автоматично створює адміністративний інтерфейс на основі моделей даних додатка, що спрощує роботу з базою даних і забезпечує швидкий доступ до адміністративних функцій.

ORM (Object-Relational Mapping). Django надає ORM-шари (Object-Relational Mapping), які дають змогу розробникам працювати з базами даних з використанням об'єктно-орієнтованого підходу. ORM в Django дає змогу створювати моделі даних, які можуть бути перетворені в таблиці в базі даних. Це спрощує взаємодію з базою даних і дає змогу розробникам працювати з даними, використовуючи об'єкти і методи, замість написання складних SQL-запитів.

Обробка форм. Django надає потужні інструменти для обробки та валідації форм на стороні сервера. Розробники можуть визначати форми за допомогою простого синтаксису Django, і фреймворк автоматично обробляє надсилання

даних форми, валідацію та відображення помилок. Це спрощує створення та обробку форм, забезпечуючи безпеку та цілісність даних.

Робота з URL-маршрутизацією. Django пропонує гнучку і потужну систему URL-маршрутизації, яка дає змогу розробникам визначати, які URL-запити мають бути оброблені якими поданнями (views). За допомогою URL-маршрутизації в Django розробники можуть легко налаштовувати шляхи та параметри запитів, а також обробляти динамічні URL-адреси.

Безпека та аутентифікація. Django надає набір інструментів для забезпечення безпеки веб-додатків. Це охоплює захист від атак CSRF (Cross-Site Request Forgery), автентифікацію користувачів, управління дозволами і контроль доступу. Django також пропонує інтеграцію з різними методами аутентифікації, включно з базовою аутентифікацією, аутентифікацією через соціальні мережі тощо.

Розширюваність і спільнота. Django є модульним і розширюваним фреймворком. Існує безліч розширень і сторонніх пакетів, які розширюють функціональність Django і спрощують розробку. Django також має активну спільноту розробників, які діляться своїм досвідом, пропонують рішення і підтримують фреймворк.

Django є потужним інструментом для розробки сучасних веб-додатків. Його функціональність, гнучкість і простота у використанні роблять його ідеальним вибором для розробників, які прагнуть створювати масштабовані та безпечні додатки. Завдяки вбудованим функціям, як-от адміністративна панель, ORM і обробка форм, Django прискорює процес розробки та підвищує продуктивність. Зі зростанням популярності Django очікується подальший розвиток і розширення його можливостей у майбутньому.

Спочатку розроблена для редакції веб-фрейм Django надає розробникам інфраструктуру, необхідну для швидкого написання веб-додатків, керованих базою даних, без необхідності починати кодування з нуля. Це потужний інструмент для прискорення загальних завдань веб-розробки, зберігаючи дизайн чистим і прагматичним.

За своєю суттю, Django є високорівневою колекцією бібліотек, написаною мовою Python високого рівня, з відкритим кодом, MVC, і однією з найпопулярніших серверних веб-платформ. Це девіз: "Не повторюйся". Як і Python, він підкреслює ефективність, дозволяючи вам робити якнайбільше з мінімальним кодуванням.

Крім того, Django також швидкий, масштабований і зрілий, з величезним співтовариством розробників та надійним набором вбудованих компонентів. На стороні даних Django може читати або генерувати екземпляри даних XML або JSON і працює в режимі сну з реляційними системами управління базами даних, такими як PostgreSQL, MySQL, SQLite і Oracle. Коли настає час розгортання, Django повністю підтримується хмарною платформою Heroku та AWS Elastic Beanstalk.

Відсутність "чарівництва" в Django означає, що ви можете запустити пошук по всьому проекту і завжди знаходити потрібний вам код. Він менш упертий і більш декларативний, що зрештою буде кращим за розробника.

Ось кілька основних функцій, які роблять його придатним для швидкої розробки веб-застосунків:

Безкоштовний, багатий на API. За допомогою простої команди Django автоматично генерує Python API на основі ваших моделей, не вимагаючи додаткового кодування.

Автоматичне створення таблиці бази даних. Команда migrate - це елегантна та відмінна риса Django, яка дивиться на всі ваші моделі та автоматично створює таблиці у вашій базі даних для тих, які ще не існують.

Генератор динамічного адміністрування інтерфейсу. Ідея тут полягає в тому, що замість вимагати від вас створення всього веб-сайту, перш ніж ви почнете заповнювати його інформацією, Django швидко і просто налаштовує адміністративний сайт, який дозволяє учасникам завантажувати базу даних, поки розробники малюють Як відображати цей контент .

Синдикаційний корм. Django також дозволяє вам швидко та легко створювати RSS та Atom-канали, створюючи простий клас Python. Ці канали

можуть бути простими (наприклад, типовий канал для останніх повідомлень) або більш спеціалізованими (наприклад, всі повідомлення в певній категорії або з певним ключовим словом). Потужна структура кеш для динамічних веб-сайтів. Ця система дозволяє скоротити дорогі обчислення, кешуючи динамічні сторінки. Тут є кілька рівнів деталізації: ви можете кешувати окремі сторінки або найдорожчі види для створення.

Потужна вбудована система шаблонів. Шаблони Django засновані на системі спадкування (аналогічно до об'єктно-орієнтованого програмування). Ці шаблони дозволяють Django динамічно генерувати HTML, одночасно усуваючи надмірність.

Проста міграція бази даних. Просте та впорядковане керування робочим процесом бази даних за допомогою перенесення Django. Керування версіями здійснюється за допомогою імен переналаштування, а Django пропонує безліч варіантів злиття версій або дозволяє розробникам вносити зміни.

Функції безпеки. Django має інструменти для захисту від XSS-атак, CSRF-атак, SQL-ін'єкцій, clickjacking і т. д. Це також дозволяє вам легко включити SSL / HTTPS та перевірку заголовка хоста.

Інші корисні доповнення. Пов'язані доповнення включають файли Sitemap, коментарі, автентифікацію користувачів, RSS-канали і т.д.

3.1.3 Вибір

Правда полягає в тому, що основна відмінність між структурами – це мови програмування, які вони використовують, підхід до мови та філософія. Python «Має бути один очевидний спосіб робити речі» суперечить гнучкості, свободі та силі вибору Ruby.

Синтаксис Python, який робить його відповідним для більшості довгострокових проєктів. Читання коду також спрощує інтеграцію новачків у команду розробників.

Python також дуже хороший для прототипування. Ви можете реалізувати нову ідею за допомогою Python. Якщо ідея життєздатна і, скажімо, ваш додаток вимагає високої продуктивності, ідея може бути повторно реалізована

компільованою високопродуктивною мовою програмування, такою як C++.

Рубі особлива мова. Я б назвав його екзотичним та витонченим. Не дивно, що мова була винайдена японцями. У цих рядках є щось унікальне, що викликає інтерес. Хоча код не так читаємо, як Python, синтаксис, як і раніше, дуже простий для розуміння та вивчення.

Ruby все є об'єктом. Кожен тип даних належить до певного класу із вбудованими методами. Те саме стосується Python, однак є кілька мов програмування, де базові типи даних не представлені об'єктами.

Інша популярна особливість Ruby називається "блоки". Використовуючи блоки, ви можете змінити будь-який метод (навіть вбудований) з закриттям користувача.

Ruby on Rails та Django – це веб-фреймворки Model-View-Controller. Цей шаблон відокремлює проблеми розвитку у трьох блоках.

Ruby on Rails і Django впорядковують файли всередині проекту.

Іншому. Уявіть, що ми маємо веб-додаток Rails з ім'ям `my_taxi`. Коли розробник додає новий компонент, наприклад, «рахунки» або «платежі», він повинен перейти в певне місце.

Django використовує окремі програми для кожного компонента. Кожна програма містить власні класи моделей, уявлень та шаблонів. Це може бути зручно, коли проект збільшується у розмірах.

Обидва Ruby on Rails та Django мають так звані «чарівні» або готові рішення. Однак магія RoP набагато перевершує всі інші.

Коли розробник rails створює контролер, його методи можуть автоматично звертатися до змінних та успадковувати методи іншого класу. Ані ці класи, ані змінні не визначені очевидно. Ruby on Rails просто додає їх автоматично. Він також автоматично встановлює маршрути всередині MVC.

Розробник Django/Python, навпаки, повинен явно визначати змінні та класи спадкування. Вони повинні використовувати регулярні вирази (щось на кшталт цього «`^[AZ]+[az\g]$`») для визначення маршрутів та URL-адрес. Розробники Django повинні повністю зрозуміти, як все працює за лаштунками, щоб

організувати код.

Обидва фреймворки засновані на скриптових мовах, що інтерпретуються. Це призводить до майже такої ж продуктивності. Ранні версії Ruby, такі як 1.8 або 1.9.x, відрізнялися низькою продуктивністю.

Python Django та Ruby on Rails виглядають як близнюки від різних батьків. У них різна індивідуальність та філософія, але коли справа доходить до роботи, вони діють однаково. Обидва є чудовими структурами, які можуть долати подібні завдання, використовуючи їх унікальні підходи.

Грунтуючись на всьому сказаному вище, був зроблений вибір на користь Rails. Ключові фактори: швидка розробка, мова, безліч готових рішень, щоб не винаходити велосипед.

3.2 Бот Microsoft bot framework

Bot Framework – середовище розробки з відкритим вихідним кодом. Орієнтована створення програм автоматичної взаємодії з користувачами через текстові повідомлення, SMS, Skype, Slack, пошту Office 365 та інші сервіси.

За допомогою Bot Framework розробники можуть створювати роботів, що діють за деякими шаблонами, що навчаються роботів. При виникненні проблем із розбором запиту програмою, бот попередить про проблему запропонує зачекати на живого оператора, який отримає запис попередньої бесіди.

Для роботи з ботами використовуються такі ключові елементи: Bot Builder SDK (доступний для C# і Node.js) створення ключової функціональності робота. Він заснований на WebAPI, і визначає протокол спілкування робота з зовнішнім світом. В рамках SDK є емулятор, що дозволяє налагоджувати ботів, а також набір класів для спрощення реалізації деяких ключових абстракцій, таких як тривалі діалоги зі станом.

Cognitive Services (раніше відомі як LUIS, складова частина Project Oxford)

дозволяють спростити аналіз природної мови та виділення сенсу з текстових речень. За допомогою веб-інтерфейсу ми можемо визначити основні синтаксичні конструкції, і автоматично привласнити їм відповідні наміри користувача (intents), на які потім реагуватиме бот.

Bot Connector дозволяє прив'язати наш бот до одного чи кількох каналам спілкування, таким як Skype, Slack, Telegram та ін.

Особливих конкурентів у даного рішення не існує, тому що Microsoft не скупився на багатоплатформенність і активно займається підтримкою та просування даного сервісу.

3.3 Хмарні платформи

Amazon Web Services (AWS): Вивчення та застосування хмарної платформи для розробки та розгортання масштабованих додатків.

Amazon Web Services (AWS) - це провідна хмарна платформа, що надається компанією Amazon. Вона пропонує широкий спектр сервісів та інфраструктури, що дають змогу організаціям розробляти, розгортати та керувати своїми додатками та інфраструктурою в хмарі. У цій статті ми розглянемо основні компоненти та сервіси AWS, їхні переваги та застосування в різних сферах.

Загальні принципи та переваги AWS. AWS пропонує низку загальних принципів і переваг, які роблять його привабливим для організацій. Масштабованість: AWS дає змогу масштабувати ресурси в хмарі залежно від потреб бізнесу. Ви можете масштабувати додатки та інфраструктуру вручну або автоматично, що забезпечує гнучкість та економію витрат.

Гнучкість і вибір: AWS пропонує широкий спектр сервісів, даючи змогу вибрати відповідні для ваших потреб рішення. Ви можете використовувати тільки ті сервіси, які вам необхідні, і налаштовувати їх відповідно до ваших вимог.

Безпека: AWS забезпечує високий рівень безпеки та конфіденційності даних.

Він пропонує інструменти та сервіси для захисту даних, контролю доступу та забезпечення дотримання вимог до безпеки.

Глобальна інфраструктура: AWS має глобальну інфраструктуру, яка дає змогу розгортати додатки та зберігати дані ближче до кінцевих користувачів. Це підвищує продуктивність і зменшує затримку для користувачів із різних регіонів.

Обчислювальні сервіси AWS. AWS пропонує широкий спектр сервісів для обробки та виконання обчислювальних завдань.

Amazon Elastic Compute Cloud (EC2): EC2 дає вам змогу орендувати віртуальні сервери в хмарі та запускати на них свої додатки. Ви можете масштабувати сервери вручну або автоматично, залежно від навантаження. AWS Lambda: Lambda є серверлес-сервісом, який дає змогу виконувати код без необхідності управління інфраструктурою серверів. Ви платите тільки за фактичний час виконання вашого коду. Amazon Elastic Container Service (ECS): ECS дає змогу запускати та керувати контейнерами з використанням Docker. Він забезпечує високу масштабованість і спрощує управління контейнеризованими додатками.

Amazon Elastic Kubernetes Service (EKS): EKS надає повністю керований сервіс Kubernetes, який спрощує розгортання, управління і масштабування додатків, заснованих на Kubernetes.

Сервіси зберігання і баз даних AWS. AWS пропонує різноманітні сервіси для зберігання та управління даними. Amazon Simple Storage Service (S3): S3 є об'єктним сховищем, яке дає змогу вам зберігати та витягувати будь-які обсяги даних. Він забезпечує високу доступність, надійність і масштабованість.

Amazon Elastic Block Store (EBS): EBS надає блокове сховище для використання з віртуальними серверами EC2. Він забезпечує постійне та надійне зберігання даних.

Amazon Relational Database Service (RDS): RDS надає керовані реляційні бази даних, включно з MySQL, PostgreSQL, Oracle та іншими. Він полегшує розгортання, масштабування та управління базами даних.

Amazon DynamoDB: DynamoDB є повністю керованою NoSQL базою даних,

яка забезпечує високу продуктивність і масштабованість. Вона ідеально підходить для додатків, які потребують швидкого доступу до даних.

Сервіси аналітики та машинного навчання. AWS пропонує безліч сервісів для аналізу даних і машинного навчання. Amazon Athena: Athena дає змогу вам виконати аналітичні запити на даних у S3 без необхідності попереднього завантаження даних у базу даних.

Amazon Redshift: Redshift - це керований сервіс сховища даних, який забезпечує високу продуктивність і масштабованість для аналізу великих обсягів даних.

Amazon EMR: EMR надає кероване середовище для обробки та аналізу великих даних з використанням Apache Hadoop і Apache Spark.

Amazon SageMaker: SageMaker - це повноцінна платформа для розробки, навчання і розгортання моделей машинного навчання. Він полегшує процес створення та застосування моделей машинного навчання.

Мережеві та безпекові сервіси AWS. AWS надає сервіси для налаштування мережі та забезпечення безпеки ваших додатків. Amazon Virtual Private Cloud (VPC): VPC дає змогу створити ізольовану віртуальну мережу в хмарі, в якій можна налаштовувати підмережі, маршрутизацію та доступ до ресурсів.

AWS Direct Connect: Direct Connect забезпечує пряме фізичне з'єднання між вашими локальними мережами та інфраструктурою AWS, забезпечуючи більш надійне і швидке з'єднання.

AWS Identity and Access Management (IAM): IAM дає змогу керувати доступом і авторизацією користувачів, груп і ролей в AWS. Він забезпечує гранулярний контроль доступу до ресурсів.

AWS Firewall Manager: Firewall Manager надає централізоване управління і контроль політик безпеки для ваших додатків і ресурсів AWS.

Потрібна платформа, яка могла б забезпечити прості у розробці запровадження рішення прямо з регулярного REST-сервісу, глибокі аналітичні можливості, черги повідомлень та особливості IoT. У цьому контексті Google відставав від AWS та Azure. Хоча Google відмінно підходить для хостингу

загальних програм та аналітики, потрібно зробити багато зв'язків, щоб пошити все разом, щоб задовольнити наші потреби.

Azure також надає умови, за яких шлюз може бути вашим власним центром обробки даних. Тільки Azure та Google дали потоки або конвеєри, які допоможуть у легкій інтеграції. У Google єдиним рекомендованим способом підключення був pub/sub, у той час як в Azure та AWS цим не обмежувалося і забезпечувало найкращі параметри. Загалом Azure став переможцем у цьому розділі, і AWS далеко позаду. Також я помітив, що тільки Azure та AWS забезпечували обмін повідомленнями «хмара-пристрій».

Для HTTP, існуючих API та мов виявилось достатньо. Google надавав підтримку SDK усіма мовами, але тільки для протоколу gRPC. Таким чином, у цьому критерії, AWS та Azure були рівні.

Крім обміну повідомленнями IoT, ми також маємо кілька варіантів використання, які строго не залежать від IoT (наприклад, реєстрація користувача тощо).

Для вирішення потрібно було подивитися ціни для різних областей, таких як база даних, вартість часу виконання функцій, потокова аналітика. Потрібно подивитися на ціни для різних обсягів даних та швидкості їх надходження (які залежать від номера пристроїв). Підійшовши до ціни, стало зрозуміло, що ціна Google значно вища. Тому Google було виключено. При переході на споживчі повідомлення вартість AWS та Azure залежить від кількості повідомлень. Але прийом - це розмір повідомлення. В AWS розмір повідомлення дорівнює 1 КБ, а в Azure – 4 КБ. Наприклад, якщо ви надсилаєте повідомлення з 8 Кб, в AWS воно буде оброблятися як вісім повідомлень, а Azure це буде два повідомлення. Тому, коли ви досягаєте ціноутворення, це потрібно враховувати під час виборів відповідного плану.

Розглядаючи всі аспекти, було обрано Azure як платформу IoT. Наголос був зроблений на те, що на відміну від AWS, продукт Microsoft відмінно підходить для використання по http протоколу без додаткових сертифікатів.

3.4 Висновки до розділу

У цьому розділі було розглянуто і проаналізовано як програмні інструменти як фреймворків розробки веб-частини, і обраний одне із найпопулярніших і ґрунтовних продуктів до створення бота. Наприкінці також було визначено хмарну платформу. Отже, стек технологій має такий вигляд: Ruby on Rails + Microsoft bot framework + Microsoft Azure.

4 РОЗРОБКА ПРОГРАМНОЇ ТА АПАРАТНОЇ ЧАСТИНИ

4.1 Апаратна частина

Для того щоб було що передавати на сервер, потрібно спочатку зібрати сирі дані в аналоговому вигляді і обробити їх у цифровій. Наріжним каменем стало завдання про те, скільки «наповнювача» зараз знаходиться в контейнері. Було висунуто 3 гіпотези/методи: датчик тиску/сили, оптичний датчик порожнечі, ехолотатор. Автором був проведений аналіз і там, де це потрібно, поставлено емпіричний досвід.

Датчик зусилля – це перетворювач, який трансформує силу чи тиск у цифровий сигнал. Величина цієї електричної потужності прямо пропорційна силі, що додається. Сенсор має тензодатчик, який деформується, коли на нього накладається тиск. А потім тензодатчик генерує електричний сигнал при зміні, тому що його ефективний опір змінюється під час деформації. Конструкція зазвичай складається з чотирьох тензодатчиків у конфігурації мосту Уітстона, що поставляється в різних діапазонах, таких як 5 кг, 10 кг, 100 кг і більше, тут була використана модель, яка може важити до 40 кг. Тепер електричні сигнали, що генеруються датчиком зусилля, знаходяться у кількох мілівольтах, тому вони повинні бути додатково посилені деяким перетворювачем, і, отже, це буде датчик

HX711. Модуль датчика HX711 має чіп HX711, який є 24-канальним високоточним аналого-цифровим перетворювачем. HX711 має два аналогових вхідних канали, і щоб було можна отримати коефіцієнт посилення до 128, шляхом програмування цих каналів. Таким чином, модуль HX711 посилює низький електричний вихід тензодатчиків, а потім цей посилений і перетворений у цифровому вигляді сигнал подається Arduino для отримання ваги.

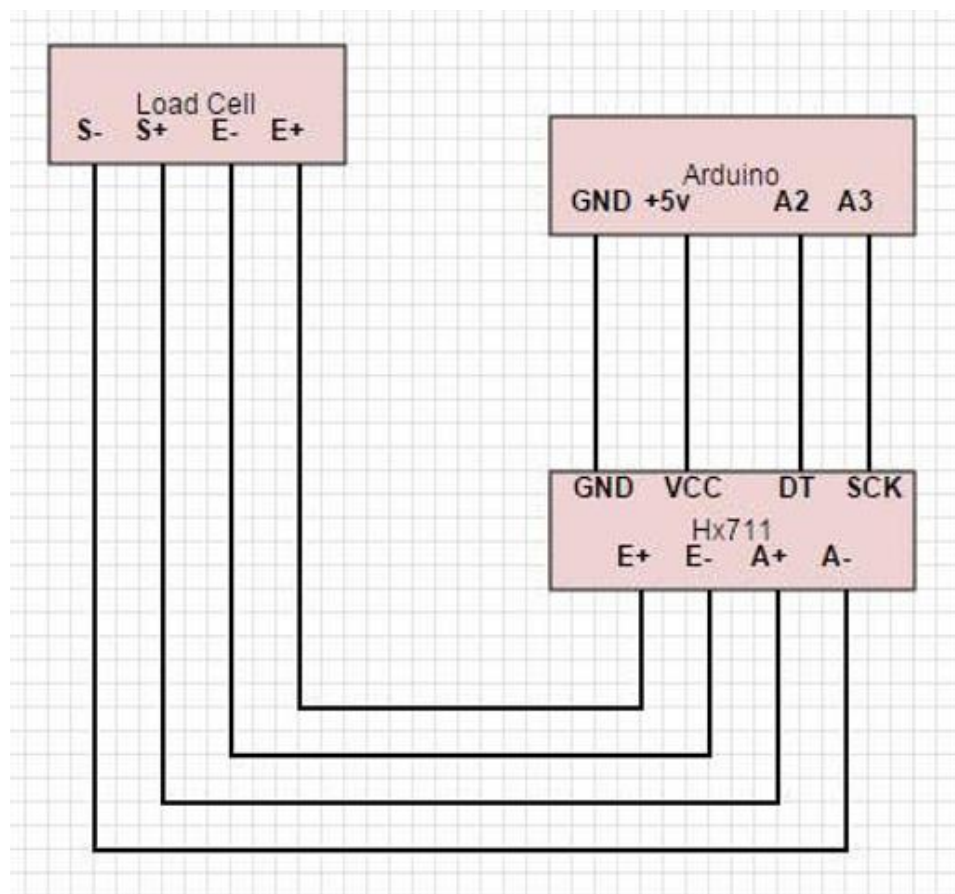


Рисунок 4.1 - Схема з'єднання

Це все призводить до того, що потрібно було створити платформу за принципом ваги для вимірювання сигналу. Подібні датчики показали себе неефективними через невелику чутливість, втрати невеликих даних, супутніх втрат в образі додаткових конструкцій. Так це виглядало:



Рисунок 4.2 - Ітерація 1

Другим варіантом стало оптичний датчик порожнечі Sharp GP2Y0A21YK0F- працює з використанням спеціального світлочутливого сенсора для визначення довжини хвилі вибраного світла в інфрачервоному спектрі. Використовуючи світлодіод, який випромінює світло на тій же довжині хвилі, що і датчик, можна побачити інтенсивність отриманого світла. Коли об'єкт близький до датчика світло від світлодіода відскакує від об'єкта і потрапляє на датчик освітленості. Це призводить до великого стрибка інтенсивності, який, як відомо, може бути виявлений з використанням порога.

Оскільки датчик працює, шукаючи відбите світло, варто мати датчик, який може повернути значення відбитого світла. Цей тип датчика може бути використаний для вимірювання яскравості об'єкта. Це корисно для таких завдань, як відстеження відстані до об'єкта. Що чудово підходить для поставленого завдання. Вираховуватиметься шлях, обчислюватиметься відстань і передаватиметься сигнал, при досягненні критичної позначки

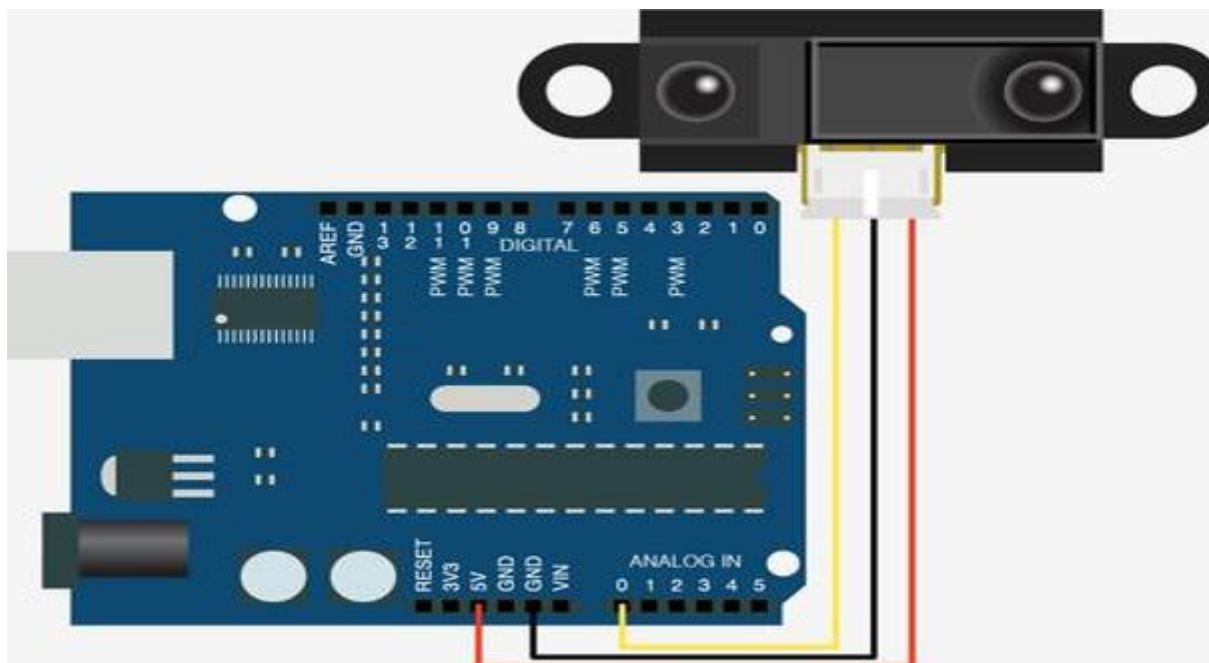


Рисунок 4.3 - Схема підключення ІЧ-датчика

В програмі Arduino дані з аналогового входу зчитуються за допомогою стандартної функції `analogRead`. Діапазон вхідної напруги від 0 до 5 В проектується на діапазон значень від 0 до 1023. Щоб дізнатися напругу на виході датчика, необхідно дані АЦП помножити на 0,0048828125 (5 В/1024). Оскільки відстань нелінійна, використовується формула для випрямлення кривої: $S = (0,0001831 \times A) - 0,0030927$, де S – відстань в умовних одиницях, A – значення на виході 10-бітного АЦП мікроконтролера. При зміні відстані від 80 до 10 см значення з виходу АЦП змінюватимуться від 70 до 475. У результаті підрахунків за наведеною формулою відстань в умовних одиницях змінюється від 0,011 до 0,085. [12]



Рисунок 4.4 - Ітерація 2

Оскільки останній варіант повністю влаштував автора, він вважав, що дослідження цьому можна закінчити. Далі наведено короткі теоретичні відомості щодо роботи з датчиком ехолокації. Він випускає ультразвук із частотою 40 000 Гц, який проходить повітрям, і якщо на його шляху є об'єкт або перешкода, він повернеться до модуля. Беручи до уваги час дії та швидкість звуку. Наприклад, якщо об'єкт знаходиться на відстані 10 см від датчика, а швидкість звуку дорівнює 340 м/с або 0,03 см/мкс, звукова хвиля має пройти близько 294 секунд. Але те, що повернеться, буде вдвічі більшим за це число, тому що звукова хвиля повинна рухатися вперед і відскакувати назад. Таким чином, щоб отримати відстань см, потрібно помножити отримане значення часу руху від ехо-сигналу на 0,034 і

розділити його на 2.

Оскільки людина, яка обслуговує пристрій, повинна мати можливість поповнювати контейнер, потрібно припиняти роботу алгоритму на цей момент, щоб дані не зчитувалися вхолосту. Оскільки, за вибором попереднього параграфа, були встановлені датчики оптичної порожнечі під кришки контейнерів, відповідно, в момент зняття однієї з кришок, має відбуватися переривання. Було висунуто на розгляд два методи: зупиняти за допомогою кнопки або за допомогою магнітних датчиків.

Кнопка - додається до схеми, підтягується резисторами до певного піна, і стан зчитується, якщо є напруга, значить зупиняється алгоритм, ні – продовжує роботу. Мінусом цього методу стала, знов-таки, людина, яка може забути натиснути на неї, не дивитися на світлодіоди, що сповіщають про роботу пристрою.

Магнітний датчик - який зазвичай використовується в дверних сигналізаціях, працює за тим же методом, якщо є напруга - значить, дані зчитуються, ні - не зчитуються. Головною відмінністю є відсутність людського фактора, співробітник відкриває кришку, все відбувається саме. Вибір зроблено на користь останнього.

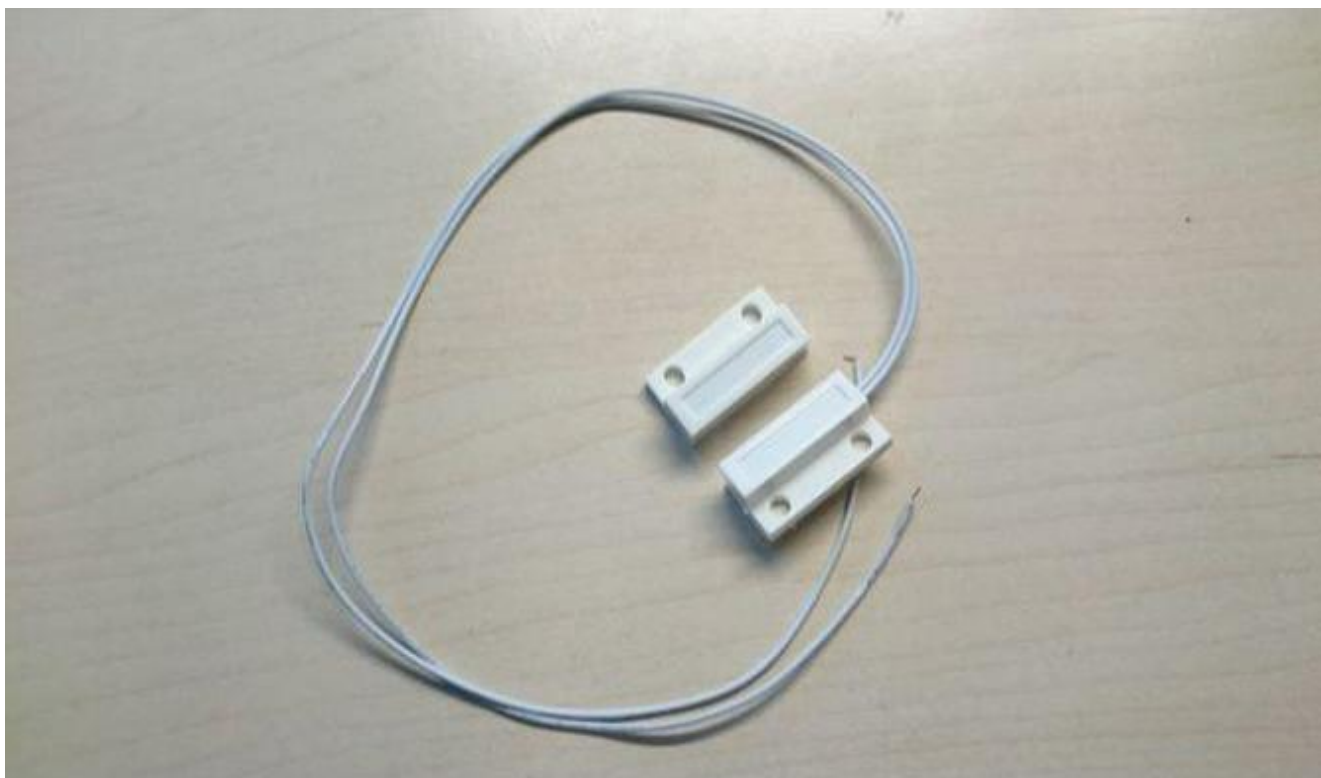


Рисунок 4.4 - Магнітний датчик контакту

Було обрано датчик gsm Sim800, який здатний передавати інформацію за допомогою http запитів на сервер. Це звільняє клієнта від потреби мати свою точку доступу до інтернету. Послідовний інтерфейс UART SIM800 спрощує для будь-якої вбудованої системи додавання доступу до мережі та підключення до хмари. Зазвичай для підключення модуля Arduino потрібно 5 проводів, і вони підключають контакти VCC / GND / Rx / Tx / Reset. Модуль VCC модуля не може безпосередньо підключатися до модуля Arduino, оскільки SIM800 вимагає робочої напруги 3,7-4,2 В. Щоб почати передавати дані потрібно перевести модуль в режим GSM-відправки, використовуючи наступні AT-команди:

```
"AT+SAPBR=3,1,\"CTYPE\",\"GPRS\",//Установка налаштувань
підключення"AT+SAPBR=3,1,\"APN\",\"internet.beeline.ru\",
"AT+SAPBR=3,1,\"USER\",\"beeline\",    "AT    +SAPBR=3,1,"PWD","beeline",
"AT+SAPBR=1,1",//Встановлюємо GPRS з'єднання "AT+SAPBR=2,1",
"AT+HTTPINIT", //Ініціалізація http сервісу "AT+HTTPPARA=\"CID\",1"
//Установка CID параметра для http
```

Далі існує таймер, який викликається раз на хвилину і відправляє дані на

сервер, наступними командами:

```
AT+HTTPPARA="URL"
```

```
AT+HTTPDATA=26,1000
```

```
AT+HTTPACTION=1
```

```
AT+HTTPTERM
```

```
AT+CIPSHUT
```

```
AT+SAPBR=0,1
```

```
AT+CGATT=1
```

4.2 Програмна частина

Передача даних відбувається таким чином: спочатку сира інформація виходить з датчиків та обробляється на Arduino, перетворюючи аналоговий сигнал на цифровий за допомогою формул. Далі, щохвилини відправляється сигнал (http-запит) на балансир, що знаходиться на платформі Azure, там сигнал фільтрується і доставляється до web-частини, що також знаходиться на продукті Microsoft, де дані візуалізуються. Паралельно існує інший потік: з телеграм робота відправляється запит на відображення статусу контейнера, його заповненості, знову ж таки на балансир, далі за схемою вище.

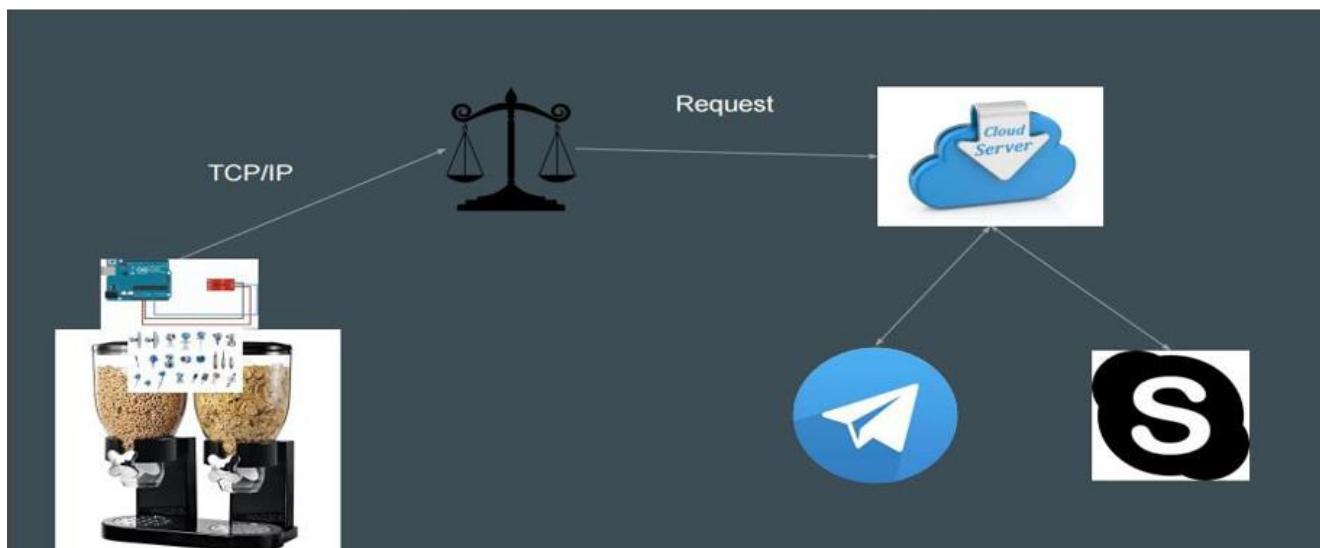


Рисунок 4.5 - Схема зв'язку

Балансування навантаження – це розподіл робочого навантаження на багатьох вузлах. В індустрії веб-хостингу він зазвичай використовується для балансування трафіку HTTP на декількох серверах, що діють як веб-інтерфейс.

Балансувальник навантаження дозволяє користувачам інтелектуально розподіляти трафік на одну IP-адресу на будь-якій кількості серверів за допомогою декількох різних протоколів. Це означає, що навантаження обробки може бути розподілене між багатьма вузлами, а не обмежуватися одним сервером, збільшуючи продуктивність у періоди високої активності. Це підвищує надійність веб-програми та дозволяє створювати програму з надлишковим обліком. Якщо один із вузлів сервера виходить з ладу, трафік програмно розповсюджується на інші вузли без переривання обслуговування. В даному випадку, оскільки є тільки один пристрій, мається на увазі, що на балансірі існує додатковий фільтр запитів, і його основна мета відокремити від прямого доступу до мережі інтернет-сервер з додатком. Використовується алгоритм Round robin:

Нехай є N об'єктів, здатних виконати задану дію, M завдань, які мають бути виконані цими об'єктами. Мається на увазі, що об'єкти n рівні за своїми властивостями між собою задачі m мають рівний пріоритет. Тоді перше завдання ($m = 1$) призначається для виконання першому об'єкту ($n = 1$), друге - другому і т. д., до досягнення останнього об'єкта ($m = N$). Тоді наступне завдання ($m = N+1$)

буде призначено знову першому об'єкту і т. п. Простіше кажучи, відбувається перебір виконуючих завдання об'єктів за циклом, або по колу (round), і після досягнення останнього об'єкта наступне завдання буде також призначено першому об'єкту. Розв'язання задач може бути додатково розбите на кванти часу, причому для продовження рішення в часі нумерація об'єктів (і, відповідно, призначені завдання) зсувається по колу на 1, тобто завдання першого об'єкта віддається другому, другого - третьому, і т. д. перший об'єкт отримує завдання останнього, чи звільняється прийому нової задачі.[13]

SnackCracker system Home Boxes			
Device #1			
Weight	Address	Phone	Content
	KFU	88005553535	peanut
	KFU	88005553535	nut mix
New Box			

Рисунок 4.6 – Веб-інтерфейс

Веб-частина написана з використанням Ruby on Rails та супутніх фреймворку шаблонів швидкої розробки. Сервіс розділений на дві частини: адміністратор та користувач. Адміністратор може створювати користувачів, контейнери для них, змінювати інформацію. Користувач може зареєструватися, змінювати контактні дані, переглядати стан контейнерів. У логіці існує обчислення критичного стану контейнера, після чого посилається сигнал про це в роботі користувача, використовується відсоткове співвідношення. Виходять дані з датчика, з дельти береться значення, що відповідає нинішньому статусу та переводиться у відсотки. Далі відбувається візуалізація в прогрес-барі, що і показується користувачеві. Для всього фронтенду використовується JavaScript.

4.3 Висновки розділу

У цьому розділі були розглянуті способи, використовувані у створенні устро́йства. Також проаналізовані датчики зі зняттям показань, вибір зроблено з урахуванням емпіричного досвіду. Далі представлений спосіб передачі за допомогою http-запитів. Схема роботи комплексу відбиває його суть. Описано дії, вчинені для створення прототипу. У результаті на виході є: Плата + датчики, що отримують і перетворюю сиру інформацію з аналогових в цифрові сигнали, має логіку обробки та переривання, і так само спосіб відправлення даних на сервер. На сервері ж сигнал приймає балансир, який виконують роль фільтра та обмежувача сервера програми від прямого доступу до мережі інтернет, з розподільника в коректному випадку надходить інформація на веб-морду, де користувач може ознайомитися з нею та редагувати дані. Паралельно працює бот, який звертається до балансира із запитом, та отримує відповідь у вигляді статусу контейнера.

ВИСНОВКИ

В результаті дипломної роботи було розроблено та спроектовано архітектуру апаратного та програмного комплексу під назвою «SmartContainer». Пристрій, що дозволяє автоматизувати та скоротити фінансові витрати поповнення продуктами харчування – снеками, офісні приміщення.

У процесі виконання дипломної роботи було вирішено такі завдання:

- проаналізовано існуючі вирішення проблеми та запропоновано своє;
- спроектовано архітектуру апаратної частини;
- проведено огляд відповідних технологій та друкованих плат;
- розроблено апаратну частину, включаючи програму мовою;
- спроектовано архітектуру та передачу даних серверно-програмної частини: балансир, веб-інтерфейс;
- розглянуто та проаналізовано інструменти, підходящі під виконання завдання;
- розроблено програмний комплекс;
- об'єднано в один пристрій під назвою "SmartContainer".

Подальший розвиток:

- поліпшення форм-фактору, промисловий дизайн;
- перехід від готових плат до власного складання;
- додавання датчика, що дозволяє дізнаватися про закінчення терміну придатності продукту функціоналу, аналітики;
- додавання бота;
- зміна дизайну веб-інтерфейсу та додавання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

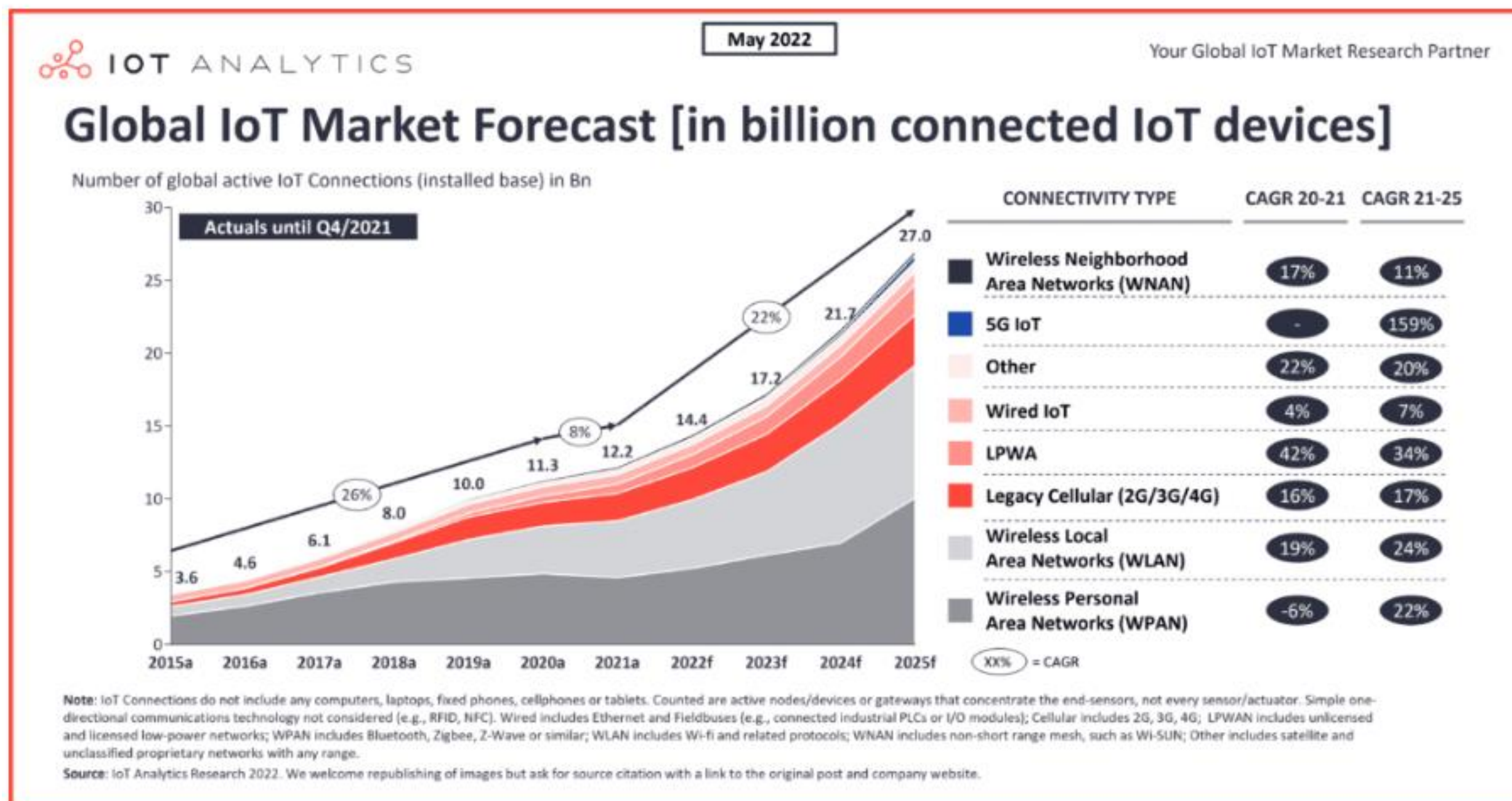
1. Стаття головного футуролога компанії Cisco Дейва Еванса
URL:<http://www.cisco.com/web/UA/about/news/2011/06262011.html>(датазвернення 28.04.2023).
2. Цикл популярних технологій 2011.
URL:<https://www.postscapes.com/internet-of-things-added-to-the-2011-hype-cycle/>(дата звернення 28.04.2023).
3. Прогноз зростання IoT аналітичної групи Juniper Research 2016.
URL:<https://www.juniperresearch.com/researchstore/key-vertical-markets/internet-of-things/consumer-industrial-public-services>. (дата звернення 01.05.2023).
4. Effects of Available Sugar on the Reproductive Fitness and Vectorial Capacity of the Malaria Vector *Anopheles gambiae* (Diptera: Culicidae) 2001 Richard E. Gary, Jr. Woodbridge A.Foster. URL:
<https://academic.oup.com/jme/article/38/1/22/1004998/Effects-of-Available-Sugar-on-the-Reproductive> (датазвернення02.05.2023).
5. Consumers Want Healthy Foods - And Will Pay More For Them. Forbes. Nancy Gagliardi 2015.
URL:<https://www.forbes.com/sites/nancygagliardi/2015/02/18/consumers-want-healthy-foods-and-will-pay-more-for-them/#68eaf0e875c5>
(датазвернення03.05.2023)
6. What Everyone Must Know About Industry 4.0. Bernard Marr. Forbes 2016
URL:<https://www.forbes.com/sites/bernardmarr/2016/06/20/what-everyone-must-know-about-industry-4-0/#b4c9ac7795f7>(дата звернення 03.05.2023).
7. Інфрачервоний датчик Sharp GP2Y0A21YK0F 2019
URL:http://electromicro.com/market/datchiki_i_sensory/infrakrasnyj_datchik_gp2y0a21yk0f/(дата звернення 20.04.2023)
8. Round-robin (алгоритм) 2011 URL:<https://wikipedia.org/wiki/Round-robin>

(дата звернення 21.05.2023)

9. Lora AT COMMANDGUIDE.REYAX TECHNOLOGY CO., LTD, 2018 року. Raspberry Pi Documentation. – Режим доступу:www.raspberrypi.org (дата звернення 21.04.2023)

10. Відкритий стандарт бездротової мережі передачі Weightless. доступу:www.weightless.org(дата звернення 21.04.2023)

Графік ринку IoT



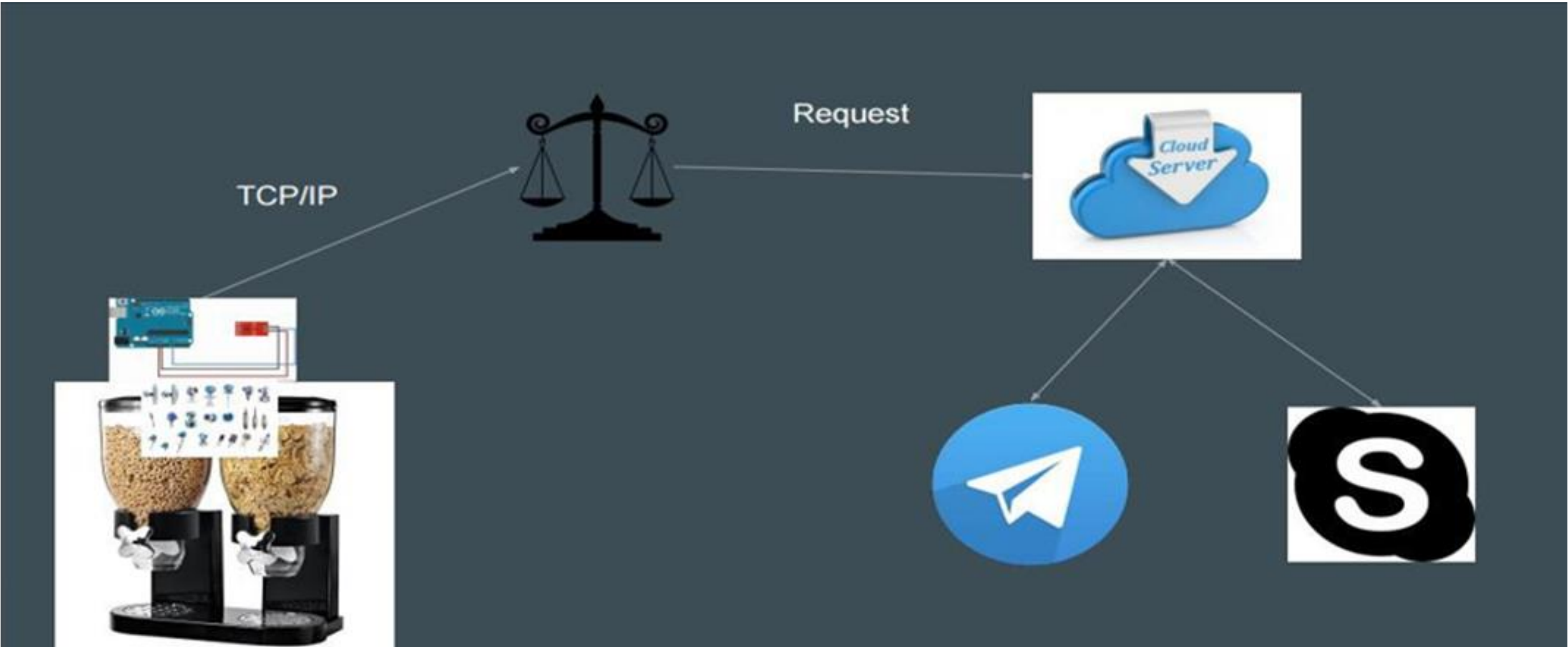
					13.02070849.51913 ПЛ1			
Зам.	Лист.	№ докум.	Підп.	Дата	Графік ринку IoT Розробка системи автоматизації рутинних офісних процесів за допомогою iot-технологій	Лист.	Маса	Масит.
Розроб.		Мінченко С.О.	М					
Перев.		Скрупський С.Ю.	С					
Т.контр.						Лист 1	Листів 1	
Н.контр.		Щербак Н.В.			НУ «Запорізька політехніка» КНТ-519			
Затв.		Кудерметов Р.К.						

Технічні характеристики плат

	Arduino Uno	Raspberry Pi Model B
Price	\$30	\$35
Size	7.6 x 1.9 x 6.4 cm	8.6cm x 5.4cm x 1.7cm
Memory	0.002MB	512MB
Clock Speed	16 MHz	700 MHz
On Board Network	None	10/100 wired Ethernet RJ45
Multitasking	No	Yes
Input voltage	7 to 12 V	5 V
Flash	32KB	SD Card (2 to 16G)
USB	One, input only	Two, peripherals OK
Operating System	None	Linux distributions
Integrated Development Environment	Arduino	Scratch, IDLE, anything with Linux support

					13.02070849.51913 ПЛ2			
					Технічні характеристики плат Розробка системи автоматизації рутинних офісних процесів за допомогою іот-технологій			
Зам.	Лист.	№ докум.	Підп.	Дата				
Розроб.		Мінченко С.О.						
Перев.		Скруський С.Ю.						
Т.контр.								
Н.контр.		Щербак Н.В.						
Затв.		Кудерметов Р.К.						
						Лист 1	Маса	Масит.
						Лист 1	Листів 1	
						НУ «Запорізька політехніка» КНТ-519		

Схема зв'язку системи



					13.02070849.51913 ПЛЗ							
					Схема зв'язку системи Розробка системи автоматизації рутинних офісних процесів за допомогою іот-технологій	Лист.			Маса		Масшт.	
Зам.	Лист.	№ докум.	Підп.	Дата								
Розроб.		Мінченко С.О.										
Перев.		Скруський С.Ю.										
Т.контр.												
						Лист 1			Листів 1			
Н.контр.		Щербак Н.В.				НУ «Запорізька політехніка» КНТ-519						
Затв.		Кудерметов Р.К.										

Хід розробки системи



Ітерація 1



Ітерація 2

					13.02070849.51913 ПЛ4			
					Хід розробки системи Розробка системи автоматизації рутинних офісних процесів за допомогою iot-технологій	Лист.	Маса	Масит.
Зам.	Лист.	№ докум.	Підп.	Дата				
Розроб.		Мінченко С.О.	<i>М</i>					
Перев.		Скруський С.Ю.	<i>С</i>					
Т.контр.						Лист 1	Листів 1	
Н.контр.		Щербак Н.В.	<i>Н</i>			НУ «Запорізька політехніка»		
Затв.		Кудерметов Р.К.	<i>Р</i>			КНТ-519		