

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«___» _____ 2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

МОНІТОРИНГ ТА ПРОГНОЗУВАННЯ ПАРАМЕТРІВ
СИСТЕМИ ОЧИЩЕННЯ ВОДИ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610201

Студент _____ А. С. Агафонов

«__» лютого 2022 р.

Консультант _____ Є. В. Сіденко
канд. техн. наук, доцент

«__» лютого 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	6
1 ОГЛЯД ОСНОВНИХ КОМПОНЕНТІВ СИСТЕМИ	8
1.1 Різноманіття фільтрів для води	8
1.2 Мова програмування	13
1.3.1 Espressif IoT Development Framework	14
1.3.2 Espruino	14
1.3.3 Lua.....	16
1.3.4 Arduino.....	17
1.3.5 Результат аналізу.....	19
1.3 Види зв'язку.....	20
1.4 Протоколи зв'язку.....	26
1.6 Моніторинг	35
Висновки до розділу 1	37
2 ПРОГНОЗУВАННЯ ПАРАМЕТРІВ	38
2.1 Що таке прогнозування, його моделі та види	38
2.2 Прогнозування часових рядів	41
2.3 Моделі прогнозування часових рядів	43
2.3.1 Naïve, Snaïve	44
2.3.2 Сезонна декомпозиція (+ будь-яка модель)	45
2.3.3 Експоненційне згладжування	47
2.3.4 ARIMA, SARIMA.....	48
2.3.5 GARCH.....	49
2.3.6 Динамічні лінійні моделі.....	49
2.3.7 TBATS	51
2.3.8 Модель Пророк.....	51
2.3.9 NNETAR.....	52
2.3.10 LSTM	53

2.3.11 Оцінка методів.....	54
2.4 Прогнозування за експоненційним згладжуванням	55
Висновки до розділу 2	57
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	58
3.1 Розробка програмного забезпечення для системи моніторингу води	58
3.2 Налаштування серверу	59
3.3 Створення ПЗ для Arduino	61
Висновки до розділу 3	67
4 МЕТОДИЧНА ЧАСТИНА	Ошибка! Закладка не определена.
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	
.....	Ошибка! Закладка не определена.
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТОК А.....	73
ДОДАТОК Б	83

ПЕРЕЛІК СКОРОЧЕНЬ

АОП – апаратна обчислювальна платформа.

ОК – одноплатний комп'ютер.

БД – база даних.

TDS - total dissolved solids.

ЗО – зворотній осмос.

USB – Universal Serial Bus.

COM – communications port.

Кафедра інтелектуальних інформаційних систем
Моніторинг та прогнозування параметрів системи очищення води

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«МОНІТОРИНГ ТА ПРОГНОЗУВАННЯ ПАРАМЕТРІВ СИСТЕМИ ОЧИЩЕННЯ ВОДИ»

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610201

Студент _____ А. С. Агафонов

«__» лютого 2022 р.

Консультант _____ Є. В. Сіденко
канд. техн. наук, доцент

«__» лютого 2022 р.

Миколаїв – 2022

ВСТУП

Вода є основою життя. Важко переоцінити значення води в забезпеченні існування життя на Землі, адже на планеті немає жодного живого організму, який би не містив у собі або не споживав воду. Для людини дуже важливо пити чисту воду – це одна з основних складових здоров'я, як і чисте повітря, здорова їжа і фізична активність. Чи траплялась у вас ситуація, коли ви відкриваєте кран і з нього витікає вода з незвичайним кольором або запахом? Якщо відкрити будь який веб-сайт з новинами, то можна бачити схожі заголовки у новин майже щодня. Такі випадки стаються через недосконалість та застарілість механізмів по очищенню води. За відкритими даними, більшість очисних споруд для води в Україні експлуатуються більше 20 років.

Найпростішим вирішенням даної ситуації для мешканців будинків з застарілим або пошкодженим водопроводом є придбання фільтра для води для додаткового очищення води із крану. Ринок фільтрів для води стрімко розвивається і на даний момент містить у собі широкий спектр рішень під будь-які потреби, та на різний бюджет.

На сьогоднішній день також стрімко розвивається тенденція так званої смартизації речей, тобто коли звичайні прибори, які на початку не були пристосовані до певних потреб, підключаються до мережі інтернет, та отримують новий функціонал. Як приклад – електронний чайник, який через смарт-розетку чи реле підмикається до живлення, і тепер їм можна керувати через інтернет – вмикати або вимикати по таймеру або натисканню кнопки, дивитись скільки електроенергії він споживає тощо.

Фільтри для води також не стали виключенням, хоча на пост-СНД ринку немає моделей, які б мали багатий функціонал моніторингу. Існує усього 3-4 моделі, які вміють показувати TDS (відносна чистота води) і температуру. Також одна з моделей вміє рахувати ресурс картриджів.

Великим недоліком даних фільтрів є їх надвисока для цього ринку ціна. Якщо звичайний фільтр зворотнього осмосу коштує приблизно \$160 - \$200, то “смарт” моделі стартують від \$700. У даній роботі буде виконано дослідження ринку фільтрів для води, буде вибрана потрібна модель, та на її базі буде створено смарт модель фільтру, який буде виводити на певний сайт показники води. Також буде проведений аналіз та прогнозування такого параметру води, як чистота (TDS), або, кажучи простіше, буде досліджено через який період часу (або через скільки літрів) потрібно міняти картриджі для фільтру.

Об’єктом дослідження є процеси очищення води.

Предметом дослідження є методи моніторингу та прогнозування

Метою роботи є моніторинг та прогнозування параметрів системи очищення води.

Для виконання поставленої мети необхідно:

- 1) проаналізувати ринок фільтрів для води;
- 2) розглянути усі доступні мікроконтролери, сформувані список вимог до них, і на основі порівняння, обрати найкращий варіант;
- 3) обрати мову програмування і середовище для програмування Arduino;
- 4) обрати типи, види і протоколи зв’язку, що відповідають заданим умовам системи;
- 5) обрати дані для моніторингу, вид і модель прогнозування;
- 6) здійснити прогнозування ресурсу картриджів для фільтру та розробити ПЗ для серверу і Arduino.

1 ОГЛЯД ОСНОВНИХ КОМПОНЕНТІВ СИСТЕМИ

1.1 Різноманіття фільтрів для води

На сьогодні, фільтри для води представлені різними виробниками. Серед них – Brita, Аквафор, Бар'єр, Aquafilter, Бриз, Ecosoft, Роса. Останні три виробники є вітчизняними. Взагалі, фільтри можна розділити на дві глобальні категорії – для побутових потреб (рис. 1.1) та корпоративні рішення (частіше за все, такі рішення використовують кав'ярні). Розглянемо побутовий сегмент. За типом, фільтр можна поділити на:

1) фільтри – кувшини (або фільтр - глечик). Фільтр, у який вставляється один картридж, зверху наливається вода, і поступово, десь за 10 хвилин, уся вода фільтрується і проливається вниз глечика;

2) настільні (або напольні) фільтри. Проточний фільтр для води, який зазвичай містить у собі два картриджі, і також часто містить власний кран для чистої води;

3) фільтри під мийку. Проточний фільтр для води, зазвичай встановлюється під мийку, містить як правило три картриджі, які підбираються у залежності від того, що потрібно фільтрувати у воді(від іржі, від накипу, від поганого запаху тощо.);

4) проточні фільтри. Виглядають як закритий стакан, який з двох сторін має різьбу. Вкручується у водопровідну трубу, і фільтрує воду. Найчастіше такі рішення використовуються для очищення води перед пральною машиною чи душем;

5) фільтри зворотнього осмосу. Складний за принципом роботи фільтр, через низьку швидкість очищення води, містить бак для води, де накопичується чиста вода(є моделі без баку, але з насосом). Принцип фільтрації наступний – вода очищується від усіх домішок до стану дестильованої води, і оскільки дестильована

вода не корисна для людини, вода мінералізується, тобто наповнюється корисними солями і мінералами до норми;

б) портативні або переносні фільтри. Це, як правило, компактні за розміром фільтри, які найчастіше використовують у походах, наприклад. Мають невеликий ресурс, та не потребують електромережі (як виключення, є фільтри для подорожування, які мають насос і працюють від 12В мережі автомобіля).

Серед усіх існуючих показників та параметрів води, було обрано найважливіші та найкорисніші для споживача. Серед них:

1) швидкість потоку води. Корисно для розуміння споживача – скільки літрів у хвилину він споживає. Також може використовуватись як індикатор того, чи забув споживач вдома закрити кран;

2) температура води. Інформація для користувача. Якщо споживач живе у будинку може допомогти виявити спад температури і своєчасно запобігти розриву водопостачання взимку;

3) тиск у системі водопостачання. Допоможе виявити аномальні значення тиску у системі водопостачання;

4) вхідний tds. Показує чистоту водопровідної води. Потрібно для наочності очищення води;

5) вихідний tds. Показує чистоту очищеної води. Потрібно для наочності очищення води;

6) сповіщення про протік води під мийкою. Допоможе своєчасно запобігти аварійній ситуації;

7) залишок ресурсу картриджів (у літрах та днях). Спрощує підрахунок ресурсу картриджів, так споживачу буде зрозуміліше, коли потрібно його міняти.



Рис.1.1. Різноманіття фільтрів для води

1.2 Апаратне забезпечення

Серцем майбутньої системи буде мікроконтролер. Мікроконтролер, або однокристальний мікрокомп'ютер - комп'ютер спеціального призначення, виготовлений з мікросхеми, що включає мікропроцесор, оперативну пам'ять і енергонезалежну пам'ять для зберігання виконуваних програм і кодів даних, порти вводу-виводу і блоки зі спеціальними функціями (лічильники, компаратори, АЦП та інші) (рис. 1.2).

Використовується для керування електронним обладнанням. По суті, це комп'ютер, здатний виконувати прості завдання. Використання однієї мікросхеми значно зменшує розмір, енергоспоживання та вартість пристроїв на основі мікроконтролерів.

Мікроконтролери можна знайти в багатьох сучасних пристроях, таких як телефони, пральні машини, і вони відповідають за роботу двигуна і гальмівних систем сучасних автомобілів, а також використовуються для створення систем

управління та систем збору інформації. Проектувати та створювати системи керування вимірювальними приладами, об'єктами та процесами на базі мікроконтролерів, які є основою систем безпеки, протипожежного захисту, рації, сигналізації тощо.

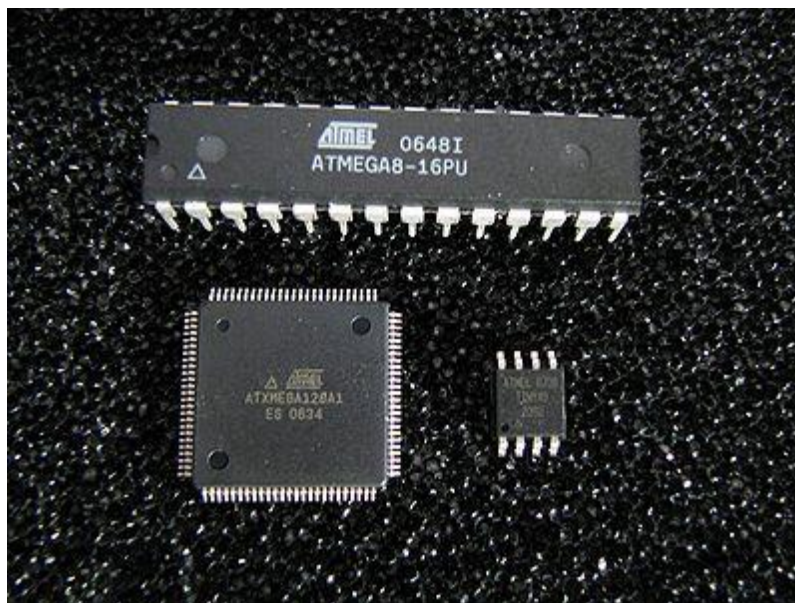


Рис. 1.2. Мікроконтролери родини AVR

Більшість процесорів у світі є мікроконтролерами. Для створення системи потрібно проаналізувати ринок мікроконтролерів, визначити їх переваги та недоліки та вибрати найкращий варіант. (Таблиця 1.1).

Таблиця 1.1

Порівняння різних моделей мікроконтролерів

Назва	Ціна	Кількість АЦП	Наявність Wi-Fi	Кількість входів/виходів
ESP8266	~\$2	1	+	17
MSP430G24	~\$2	8	-	16
nRF52832	~\$6	8	-	32

Продовж. табл. 1.1

Назва	Ціна	Кількість АЦП	Наявність Wi-Fi	Кількість входів/виходів
ATMEGA32U4-AU	~\$5	12	-	26
PIC16F877A	~\$4	14	-	40
ATtiny85	~\$2	4	-	6
ESP32	~\$3	15	+	34
ATmega328	~\$3	6	-	20
STM32F103C8T6	~\$2	10	-	32

Основними критеріями для вибору мікроконтролера є наявність 4+ аналогових входів (ADC, чи АЦП) і наявність бездротового модуля. У представленій таблиці видно, що під ці критерії підходить тільки ESP32. Саме цей мікроконтролер є найкращим варіантом для даної системи. (Рис. 1.3).

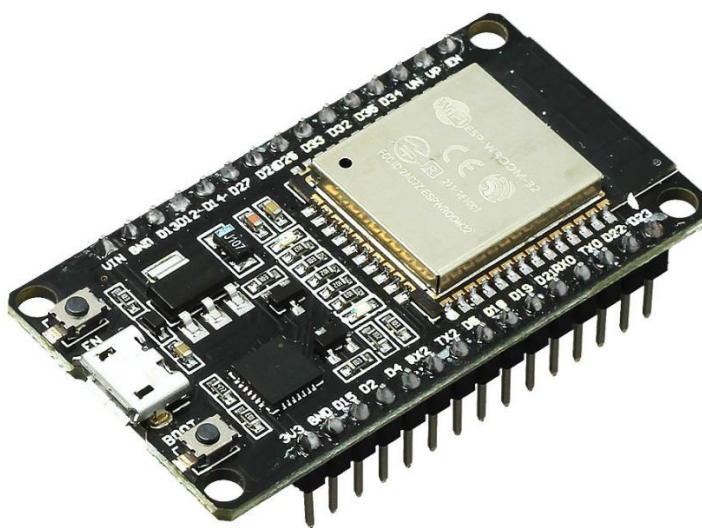


Рис. 1.3. ESP32

У даній моделі на борту вже є бездротові модулі, що відповідають за передачу даних через такі технології, як Wi-Fi та Bluetooth. Основними перевагами даного мікроконтролера є:

- 1) невисока ціна;
- 2) швидкий, порівняно з конкурентами, процесор (240 або 160 МГц);

- 3) 520 Кб вбудованої пам'яті;
- 4) 12-розрядний АЦП, що підтримує до 18 каналів;
- 5) вбудований датчик температури;
- 6) два I2S інтерфейси;
- 7) два I2C інтерфейси;
- 8) три порти UART;
- 9) наднизьке споживання електроенергії (у режимі сну – 5 мкА, у роботі – до 50 мкА).

Але, як і у будь-якого приладу, мікроконтролер також має декілька недоліків:

1) через певні інженерні недоліки, при використуванні Wi-Fi, більшість аналогових портів стає недоступними, тому нажалі із 15 портів, при роботі з Wi-Fi **можна використовувати усього 6;**

2) більшість програмних бібліотек Arduino за замовчуванням не підтримують цей мікроконтролер;

3) якщо плата буде запаяна у інтегральну схему, її неможливо буде перепрограмувати, так як при замиканні деяких портів (наприклад EN).

Тепер, коли відомо про всі недоліки та переваги даного пристрою, можна перейти до вибору потрібного ПЗ для роботи з мікроконтролером.

1.2 Мова програмування

Якщо проаналізувати перших 100 найпопулярніших мов програмування, то з'ясується, що мало мов програмування саме підходить для керування мікроконтролерами. Також потрібна мова програмування не має конфліктувати з ESP32. Розглянемо популярні мови програмування, платформи та середовища, їх переваги та недоліки.

1.3.1 Espressif IoT Development Framework

ESP-IDF (Espressif IoT Development Framework) — офіційне середовище для розробки додатків IoT на основі чіпової системи ESP32. Вперше продукт був запущений у грудні 2016 року. З тих пір програмне забезпечення ESP-IDF було оптимізовано, як це відображено в його новій версії. Після випуску версії 2.1.1 у грудні 2017 року Espressif випустив версію 3.0 ESP-IDF наприкінці квітня 2018 року та версію 3.2 рік пізніше, наприкінці квітня 2019 року.

Однією з найважливіших змін, внесених у середовище ESP-IDF версії 3.2, є підтримка уніфікованої конфігурації. Ця підтримка забезпечує розширений механізм використання різних схем транспортування та безпеки для розробників, які хочуть налаштувати свої пристрої за допомогою облікових даних Wi-Fi та/або інших налаштувань користувача. Уніфікована конфігурація ESP-IDF — це повне, готове рішення для ініціалізації мережі Wi-Fi, а також приклади додатків для iOS та Android, залежно від конкретного випадку використання. Крім того, розробники можуть розширити реалізації на стороні пристрою та мобільного додатка, щоб задовольнити власні потреби щодо надсилання додаткових даних конфігурації.

Найголовнішим недоліком даного середовища є його непопулярність, і, як наслідок, відсутність різноманіття документацій, бібліотек та прикладів робіт, що в разі ускладнює роботу. Мабуть єдиною перевагою буде гарна оптимізація та мультиплатформенність даного пакету ПЗ.

1.3.2 Espruino

Espruino — це інтерпретатор JavaScript з відкритим вихідним кодом для мікроконтролерів. Він призначений для пристроїв з невеликим об'ємом оперативної пам'яті (всього 8 Кб). Espruino був створений Гордоном Вільямсом у 2012 році, щоб зробити розробку мікроконтролерів справді мультиплатформенною. Хоча мікропрограмне забезпечення Espruino спочатку не є відкритим вихідним кодом, воно доступне для безкоштовного завантаження для

мікроконтролерів STM32. Він був запущений у 2013 році після успішної кампанії на Kickstarter з продажу плат розробників, на яких працює програмне забезпечення. Після виходу оригінальної плати Espruino з'явилося кілька нових офіційних плат розробки, включаючи невелику Espruino Pico з USB-накопичувачем, Espruino WiFi з підтримкою Wi-Fi, Puck.js з вбудованим Bluetooth і Pixl.js LCD з вбудованою в і сумісність з щитом Arduino. На додаток до офіційних плат для розробки, Espruino працює на близько 40 інших типах плат розробки, включаючи ESP8266.

Espruino надає багатий довідковий матеріал, включаючи понад 100 підручників, а також книгу «Making Things Smart», яка містить колекцію апаратних проектів, які можна створити за допомогою мікроконтролерів на базі Espruino. Першою офіційною дошкою для розробки є Original Espruino. Пізніші дошки доступні в різних формах. За оригінальним Espruino слідували Espruino Pico, Espruino WiFi, Puck.js та Pixl.js. Також є перемикач з модулем MDBT42Q Bluetooth LE для Puck.js і Pixl.js. Програми Espruino написані на JavaScript (Рис.1.4). Espruino IDE доступний як веб-додаток, Google Chrome і власна програма для Windows. Альтернативи програмуванню плати Espruino включають використання термінальної програми, такої як PuTTY у Windows.

Перевагами даного продукту є те, що він дозволяє використовувати JavaScript для розробки на Arduino. Завдяки такому підходу, усі бібліотеки, функції, класи і т.д., що існували лише для JS, тепер можна використовувати і на Arduino. Однозначно, це одна із найперспективніших ПЗ, яка має багато переваг, таких як оптимізований софт, та дуже широкий спектр бібліотек та документацій.

4) простота синтаксисів та розгортання. Вивчення мови не вимагатиме багато часу у тих, хто вже займається програмуванням. Для новачків ця мова також буде просто у вивченні;

5) програми та програми, написані на Lua споживають менше пам'яті. Що стосується недоліків, до них можна віднести те, що мова є скриптовою. Звичайно, його можна використовувати і самостійно, проте найчастіше він застосовується разом з іншими мовами програмування, наприклад С. Тому для повноцінної роботи знання ще однієї мови розробки.

1.3.4 Arduino

З моменту запуску платформи з відкритим кодом Arduino бренд утвердився в центрі великої спільноти з відкритим кодом. Екосистема Arduino складається з різноманітної комбінації апаратного та програмного забезпечення. Універсальність Arduino та його простий інтерфейс роблять його провідним вибором для широкого кола користувачів у всьому світі від любителів, дизайнерів і художників до прототипів продуктів.

Плата Arduino підключається до комп'ютера через USB, де вона підключається до середовища розробки Arduino (IDE). Користувач пише код Arduino в IDE, а потім завантажує його на мікроконтролер, який виконує код, взаємодіючи з входами та виходами, такими як датчики, двигуни та освітлення.

Як новачки, так і експерти мають доступ до безлічі безкоштовних ресурсів і матеріалів для їх підтримки. Користувачі можуть знайти інформацію про те, як налаштувати свою плату або навіть як кодувати на Arduino. Відкритий вихідний код Arduino зробив його особливо дружнім для нових і досвідчених користувачів. В Інтернеті доступні тисячі прикладів коду Arduino.

Код Arduino написаний на C++ з додаванням спеціальних методів і функцій. C++ — це доступна для читання мова програмування. Коли створюється «ескіз» (назва файлів коду Arduino), він обробляється та компілюється на машинну мову.

Інтегроване середовище розробки Arduino (IDE) є основною програмою для редагування тексту, яка використовується для програмування на Arduino. Тут буде створюватись код, перш ніж завантажуватись на плату, яку потрібно запрограмувати. Код Arduino називають ескізами .

Послідовний монітор Arduino можна відкрити, натиснувши на значок лупи у верхньому правому куті IDE або під інструментами. Послідовний монітор використовується в основному для взаємодії з платою Arduino за допомогою комп'ютера і є чудовим інструментом для моніторингу та налагодження в реальному часі. Щоб використовувати монітор, потрібно використовувати клас Serial.

За допомогою монітору COM-порту можна не тільки спостерігати за вихідними даними, а й надсилати дані до Arduino. (Рис.1.5.).

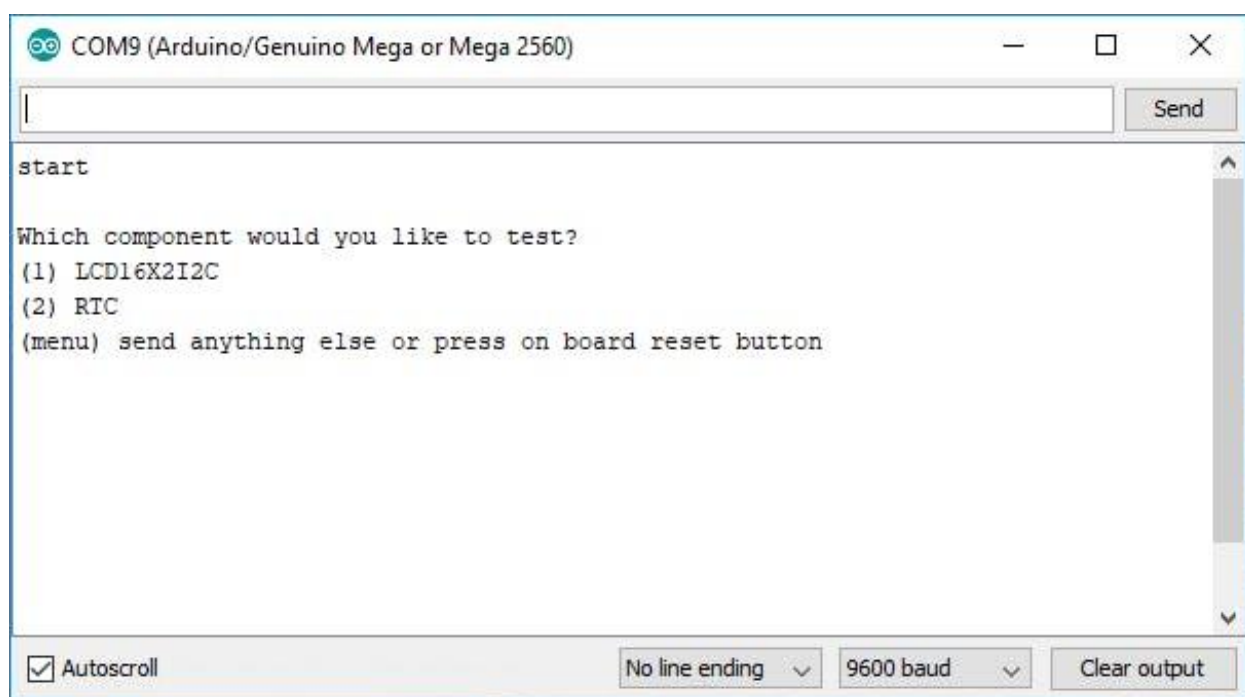


Рис. 1.5. Монітор COM-порту

Послідовний плоттер Arduino — це ще один компонент Arduino IDE, який дозволяє генерувати графік ваших послідовних даних у режимі реального часу.

Послідовний плотер значно полегшує аналіз даних за допомогою візуального дисплея. Тут можна створювати графіки, графіки від'ємних значень та проводити аналіз форми сигналу.

На відміну від інших платформ програмного програмування, Arduino не має вбудованого налагоджувача. Користувачі можуть використовувати стороннє програмне забезпечення, або вони можуть використовувати послідовний монітор для друку активних процесів Arduino для моніторингу та налагодження.

Використовуючи клас Serial, можна друкувати на моніторі послідовного зв'язку, налагоджуючи коментарі та значення змінних. На більшості моделей Arduino будуть використовуватись послідовні контакти 0 і 1, які підключені до порту USB.

1.3.5 Результат аналізу

Проаналізувавши популярні мови програмування те середовища, найкращим варіантом все ж залишається стандартна мова програмування Arduino, так як на це є декілька причин:

1) Ардуіно все ще залишається найпопулярнішою програмною платформою для апаратної платформи Ардуіно. Результатом цього є велика кількість інформації про Ардуіно саме для програмної мови Ардуіно;

2) найбільша кількість бібліотек та скетчів також відноситься до мови Arduino;

3) через таку популярність, виробники різноманітного приладдя для ардуіно (сенсори, реле, модулі тощо) зазвичай в додаток до свого пристрою додають інструкцію, у якій як приклад вказаний скетч саме на мові Arduino.

Через ці причини було вирішено створювати ядро системи саме на мові Arduino. Тепер для майбутньої системи треба обрати вид зв'язку, а за ним протокол передачі даних, який максимально підійде до даної системи.

1.3 Види зв'язку

На сьогоднішній день існує багато видів зв'язку. Так як майбутня система буде вмонтована до фільтру, то це накладує певні вимоги до типу підключення. Наприклад, використовувати Ethernet підключення на жаль не є можливим, так як для цього споживачам доведеться спеціально прокласти виту пару до кухні - під мийку, що не є задовільним. Через це необхідно розглядати тільки бездротові варіанти підключення. Для цього потрібно створити цільового споживача даної системи. Скоріш за все це сім'я, 30-50 років, у якої вдома є маршрутизатор (або роутер), який під'єднаний до мережі інтернет. Вважаючи це за стандартну ситуацію, треба розглянути усі види зв'язку та їх користь і ефективність:

GSM (Глобальна система мобільного зв'язку) — це цифрова мобільна мережа, яка широко використовується користувачами мобільних телефонів у Європі та інших частинах світу. GSM використовує різновид множинного доступу з тимчасовим розділенням (TDMA) і є найбільш широко використовуваною з трьох технологій цифрової бездротової телефонії: TDMA, GSM і множинний доступ з кодовим розподілом (CDMA). GSM оцифровує та стискає дані, а потім надсилає їх по каналу з двома іншими потоками даних користувача, кожен у своєму часовому інтервалі. Він працює в діапазоні частот 900 мегагерц (МГц) або 1800 МГц. GSM разом з іншими технологіями є частиною еволюції бездротових мобільних телекомунікацій, які включають високошвидкісні комутаційні дані (HSCSD), загальну пакетну службу радіозв'язку (GPRS), розширене середовище GSM даних (EDGE) та Universal Mobile Telecommunications Service (UMTS).

На жаль, дана технологія потребує сім-карту та специфічний модуль для неї, що негативно відобразиться на ціні. До того ж, сім-карту комусь потрібно буде обслуговувати, а це не вигідно ні виробнику, ні споживачу. Також із мінусів варто виділити періодичні проблеми із зоною покриття мережі і невисокою швидкістю, що робить цю технологію непривабливою для нашої системи.

Zigbee — це заснована на стандартах бездротова технологія, розроблена для створення недорогих, малопотужних бездротових мереж «машина-машина» (M2M) та Інтернету речей (IoT). Zigbee призначений для додатків з низькою швидкістю передачі даних і з низьким енергоспоживанням і є відкритим стандартом. Теоретично це дає змогу поєднувати реалізації від різних виробників, але на практиці продукти Zigbee були розширені та налаштовані постачальниками і, таким чином, страждають від проблем сумісності. На відміну від мереж Wi-Fi, які використовуються для підключення кінцевих точок до високошвидкісних мереж, Zigbee підтримує набагато нижчу швидкість передачі даних і використовує мережевий протокол, щоб уникнути концентраторів і створити архітектуру самовідновлення. Zigbee базується на специфікації 802.15 Асоціації стандартів Інституту електротехніки та електроніки (IEEE). Zigbee створено для мереж керування та сенсорів за бездротовим стандартом IEEE 802.15.4 для бездротових персональних мереж (WPAN). Zigbee WPAN працюють на частотах 2,4 ГГц, 900 МГц і 868 МГц. Специфікації Zigbee, які підтримується та оновлюються Zigbee Alliance, покращують стандарт IEEE 802.15.4, додаючи рівні мережі та безпеки на додаток до платформи програми. Стандарти, створені альянсом, можна використовувати для створення сумісних пропозицій від кількох постачальників. Виробники, які розробляють спеціальні програми, яким не потрібно працювати з додатками інших виробників, можуть створювати власні специфічні варіанти та розширення. На момент написання цієї статті існує три специфікації Zigbee: Zigbee PRO, Zigbee RF4CE та Zigbee IP.

Zigbee PRO прагне створити основу для IoT з функціями підтримки недорогих високонадійних мереж для зв'язку між пристроєм. Zigbee PRO також пропонує Green Power, нову функцію, яка підтримує збір енергії або пристрої з автономним живленням, які не вимагають батарей або джерела живлення змінного струму.

Zigbee RF4CE розроблено для простих, двосторонніх програм керування між пристроєм, яким не потрібні повнофункціональні функціональні можливості сітчастої мережі, запропоновані специфікацією Zigbee.

Zigbee IP оптимізує стандарт для повних бездротових мереж на основі IPv6, пропонуючи підключення до Інтернету для керування малопотужними та недорогими пристроями.

Дана бездротова технологія містить у собі багато таких переваг, як низький рівень споживання електроенергії, відносно висока швидкість, великий радіус покриття і т.д. Але, усі ці переваги перебиває один недолік – для роботи з таким стандартом потрібно також мати маршрутизатор (роутер), або бідь-який інший приймач, що буде підтримувати технологію Zigbee. Нажаль, на сьогоднішній день занадто мало людей використовують цю технологію, щоб на її основі створювати систему, тому від ZigBee доведеться відмовитись.

Ethernet – це традиційна технологія для підключення пристроїв до дротової локальної мережі (LAN) або глобальної мережі (WAN). Це дає змогу пристроям спілкуватися один з одним за допомогою протоколу, який є набором правил або загальною мовою мережі. Ethernet описує, як мережеві пристрої формують і передають дані, щоб інші пристрої в тій самій локальній мережі або мережі кампуса могли розпізнавати, отримувати та обробляти інформацію. Кабель Ethernet — це фізична закрита проводка, по якій передаються дані. Підключені пристрої, які використовують кабелі для доступу до географічно локалізованої мережі – замість бездротового з'єднання – швидше за все, використовують Ethernet. Від компаній до геймерів різноманітні кінцеві користувачі покладаються на переваги підключення Ethernet, які включають надійність та безпеку.

У порівнянні з технологією бездротової локальної мережі (WLAN), Ethernet, як правило, менш вразливий до збоїв. Він також може запропонувати більший ступінь безпеки та контролю мережі, ніж бездротові технології, оскільки пристрої повинні підключатися за допомогою фізичного кабелю. Це ускладнює для

сторонніх осіб доступ до мережевих даних або захоплення пропускну здатності для несанкціонованих пристроїв.

На сьогоднішній день, Ethernet є найбільш популярним способом з'єднання з мережею інтернет, але його недолік у тому, що до фільтру (у кухню) доведеться тягнути провод, що не є зручним. Доведеться перейти до наступної технології. **Bluetooth** – це технологія бездротового зв'язку малої дальності, яка дозволяє таким пристроям, як мобільні телефони, комп'ютери та периферійні пристрої, передавати дані або голос бездротовим способом на невеликі відстані. Метою Bluetooth є заміна кабелів, які зазвичай з'єднують пристрої, зберігаючи безпеку зв'язку між ними. Назва «Bluetooth» взята від датського короля 10-го століття на ім'я Харальд Блутуз, який, як кажуть, об'єднує розрізнені, ворогуючі регіональні фракції. Як і її тезка, технологія Bluetooth об'єднує широкий спектр пристроїв у багатьох різних галузях за допомогою єдиного стандарту зв'язку. Розроблений у 1994 році Bluetooth був призначений як бездротова заміна кабелів. Він використовує ту ж частоту 2,4 ГГц, що й деякі інші бездротові технології в домі чи офісі, наприклад, бездротові телефони та маршрутизатори Wi-Fi. Він створює бездротову мережу радіусом 10 метрів (33 фути), яка називається персональною мережею (PAN) або пікомережею, яка може підключатися до мережі від двох до восьми пристроїв. Ця мережа короткого радіусу дії дозволяє надсилати сторінку на принтер в іншій кімнаті, наприклад, без необхідності прокладати непривабливий кабель. Bluetooth споживає менше енергії та коштує менше, ніж Wi-Fi. Його низька потужність також робить його набагато менш схильним страждати від інших бездротових пристроїв у тому ж діапазоні 2,4 ГГц або створювати перешкоди. Діапазон Bluetooth і швидкість передачі зазвичай нижчі, ніж Wi-Fi (бездротова локальна мережа, яка може бути у вас вдома). Bluetooth версії 3.0 + HS — високошвидкісна технологія Bluetooth — пристрої можуть передавати дані зі швидкістю до 24 Мбіт/с, що швидше, ніж стандарт WiFi 802.11b, але повільніше, ніж стандарти Wireless-a або Wireless-g. Однак у міру розвитку технологій швидкість Bluetooth зростає.

Специфікація Bluetooth 4.0 була офіційно прийнята 6 липня 2010 року. Функції Bluetooth версії 4.0 включають низьке енергоспоживання, низьку вартість, сумісність різних виробників і розширений діапазон.

Відмітною особливістю характеристики Bluetooth 4.0 є зниження вимог до живлення; пристрої, що використовують Bluetooth версії 4.0, оптимізовані для роботи з низьким зарядом батареї та можуть працювати від невеликих батарейок типу «таблетка», відкриваючи нові можливості для бездротових технологій. Замість того, щоб боятися, що якщо залишити Bluetooth увімкненим, акумулятор розрядить ваш мобільний телефон, наприклад, ви можете залишити мобільний телефон Bluetooth версії 4.0 постійно підключеним до інших аксесуарів Bluetooth.

Ця технологія є перспективною для майбутньої системи, але вона не дає можливості завантажувати дані до мережі інтернет. Bluetooth лише дає змогу моніторити показники у в безпосередній близькості до фільтру, що позбавляє користувача таких корисних функцій, як можливість віддалено спостерігати, чи є у системі водопостачання протік, чи перепади тиску.

USB (COM) – це інтерфейс підключення і відтворення, який дозволяє комп'ютеру зв'язуватися з периферійними та іншими пристроями. Пристрої, підключені через USB, охоплюють широкий діапазон; будь-що, від клавіатур і мишок до музичних плеєрів і флеш-накопичувачів. Додаткову інформацію про ці пристрої див. у розділі USB-пристрої. USB також може передавати живлення на певні пристрої, наприклад, живлення смартфонів і планшетів і заряджання їх акумуляторів. Перший комерційний випуск універсальної послідовної шини (версія 1.0) був у січні 1996 року. Потім цей галузевий стандарт був швидко прийнятий Intel, Compaq, Microsoft та іншими компаніями.

Якщо використовувати інтерфейс USB, то Arduino доведеться підключати до комп'ютеру чи ноутбуку, що є непрактичним і незручним. Також це змусить користувача інстальювати пакет програм, налаштовувати COM інтерфейси і т.д.

Wi-Fi став повсюдно поширеною технологією в ноутбуках більше десяти років тому, і ця фраза використовується щоденно. Тепер технологію бездротової мережі можна знайти в наших годинниках, телефонах, телевізорах і розумних колонках. Що означає Wi-Fi? Філ Беланджер, член-засновник Wi-Fi Alliance, відповів на це питання так: «Wi-Fi нічого не означає. Це не аббревіатура. Тут немає сенсу». Проста істина полягає в тому, що організації потрібна була назва для свого стандарту, яку було б легше запам'ятати, ніж «Пряма послідовність IEEE 802.11b». Тому вони найняли маркетингове агентство Interbrand, щоб створити назву, і отримали на вибір 10 варіантів. Беланджер визнає, що на початку Wi-Fi організація включила тег «Стандарт бездротової вірності». Це сталося тому, що членам правління було важко уявити собі ім'я, яке нічого не означає, і вони хотіли згодом пояснити ім'я, яке було вигадано для них, але Беланджер каже, що це було помилкою, і рядок швидко відкинули. Для багатьох людей термін Wi-Fi тепер є синонімом «Інтернет». Але якщо бути точним, це просто стандарт бездротового підключення. Якщо ви використовуєте бездротовий пристрій, то Wi-Fi може забезпечувати підключення до Інтернету, але це не Інтернет. Wi-Fi, по суті, є дуже просунутим цифровим радіо, яке використовує частоти від 2 гігагерц до 5 гігагерц в електромагнітному спектрі, що є приблизно такою ж площею, що й мікрохвильові печі.

Мережі Wi-Fi не мають фізичного дротового з'єднання між відправником і одержувачем. Натомість вони функціонують, використовуючи радіочастотну технологію, частоту в межах електромагнітного спектру, пов'язану з поширенням радіохвиль. Коли радіочастотний струм подається на антену, створюється електромагнітне поле, яке потім може поширюватися в просторі. Основою будь-якої бездротової мережі є точка доступу (AP). Основним завданням точки доступу є передача бездротового сигналу, який комп'ютери можуть виявити та використовувати для встановлення з'єднання з мережею.

Щоб підключитися до точки доступу та приєднатися до бездротової мережі, комп'ютери та пристрої повинні бути оснащені адаптерами бездротової мережі.

Так як на сьогоднішній день, майже у кожної людини вдома є мережа Wi-Fi, ця технологія є найбільш підходящою для системи. Тепер, коли вибрано пристрій, мову програмування для нього, вид зв'язку, залишилось лише обрати протокол зв'язку.

1.4 Протоколи зв'язку

Розглянемо найпопулярніші протоколи і стандарти для IoT. На сьогоднішній день, можна виділити приблизно дев'ять технологій.

CoAP (Constrained Application Protocol - Протокол обмеженого застосування) - Хоча існуюча інфраструктура Інтернету є вільно доступною та придатною для використання для будь-якого пристрою IoT, вона часто виявляється занадто важкою та енерговитратною для більшості випадків використання IoT. Створений робочою групою IETF Constrained RESTful Environments та запущений у 2013 році, Constrained Application Protocol (CoAP) був розроблений для перекладу моделі HTTP, щоб її можна було використовувати в обмежувальних пристроях і мережевих середовищах.

Створений для задоволення потреб систем Інтернету речей на основі HTTP, CoAP покладається на протокол User Datagram Protocol (UDP) для встановлення безпечного зв'язку між кінцевими точками. Дозволяючи широкомовну та багатоадресну передачу, UDP може передавати дані кільком хостам, зберігаючи при цьому швидкість зв'язку та низьку пропускну здатність, що робить його гарною відповіддю для бездротових мереж, які зазвичай використовуються в середовищах M2M з обмеженими ресурсами (Рис. 1.6). Ще одна річ, яку CoAP поділяє з HTTP, — це архітектура RESTful, яка підтримує модель взаємодії запит/відповідь між кінцевими точками програми. Більше того, CoAP використовує

основні методи HTTP get, post, put і delete, завдяки яким можна уникнути неоднозначності під час взаємодії між клієнтами.

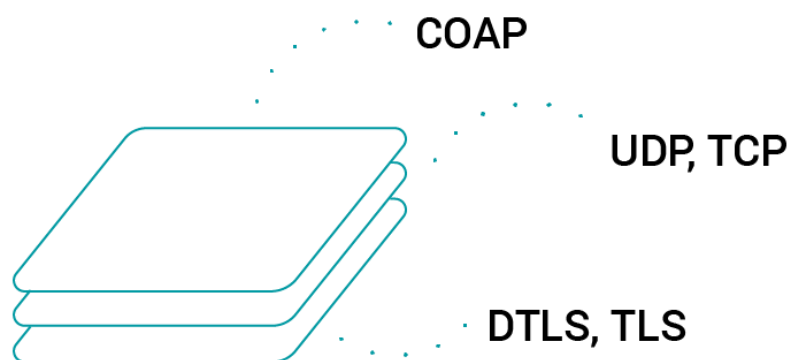


Рис. 1.6. Протокол CoAP

CoAP має функцію якості обслуговування, яка використовується для контролю надісланих повідомлень і позначення їх як «підтверджені» або «непідтверджені», відповідно, що вказує, чи повинен одержувач повернути «підтвердження» чи ні. Іншими цікавими особливостями CoAP є те, що він підтримує узгодження вмісту та механізм виявлення ресурсів. Окрім передачі даних IoT, CoAP використовує захист транспортного рівня дейтаграм (DTLS) для безпечного обміну повідомленнями на транспортному рівні. CoAP повністю задовольняє потреби надзвичайно легкого протоколу, щоб задовольнити вимоги пристроїв, що працюють від батареї або з низьким енергоспоживанням. Загалом, CoAP добре підходить, коли справа доходить до існуючих систем IoT на основі веб-сервісів.

MQTT (Message Queuing Telemetry Transport - Транспорт телеметрії черги повідомлень) - Ймовірно, найпоширеніший стандарт промислового Інтернету речей на сьогоднішній день, телеметричний транспорт черги повідомлень — це легкий протокол обміну повідомленнями типу публікації/підписки (pub/sub). Архітектура MQTT, розроблена для пристроїв, що живляться від батарей, є простою та легкою, що забезпечує низьке споживання

енергії для пристроїв. Працюючи поверх протоколу TCP/IP, він був спеціально розроблений для ненадійних комунікаційних мереж, щоб відповісти на проблему зростання кількості невеликих дешевих малопотужних об'єктів, які з'явилися в мережі в останні роки.

MQTT заснований на моделі передплатника, видавця та брокера (Рис. 1.7). У рамках моделі завдання видавця — збирати дані та надсилати інформацію передплатникам через рівень посередництва, який є брокером. Роль брокера, з іншого боку, полягає в забезпеченні безпеки шляхом перехресної перевірки авторизації видавців і передплатників. MQTT пропонує три режими досягнення цього (Якість обслуговування), завдяки яким видавець має можливість визначити якість свого повідомлення:

QoS0 (максимум один раз): найменш надійний режим, але також найшвидший. Публікацію надіслано, але підтвердження не отримано.

QoS1 (принаймні один раз): гарантує, що повідомлення буде доставлено принаймні один раз, але можуть бути отримані дублікати.

QoS2 (точно один раз): найнадійніший режим із найбільшим споживанням пропускну здатності. Дублікати контролюються, щоб повідомлення було доставлено лише один раз.

Знайшовши широке застосування в таких пристроях IoT, як електролічильники, транспортні засоби, сповіщувачі, промислове чи санітарне обладнання, MQTT добре відповідає наступним потребам:

- 1) мінімальне використання пропускну здатності;
- 2) робота через бездротові мережі;
- 3) низьке споживання енергії;
- 4) хороша надійність при необхідності;
- 5) мало ресурсів обробки та пам'яті.

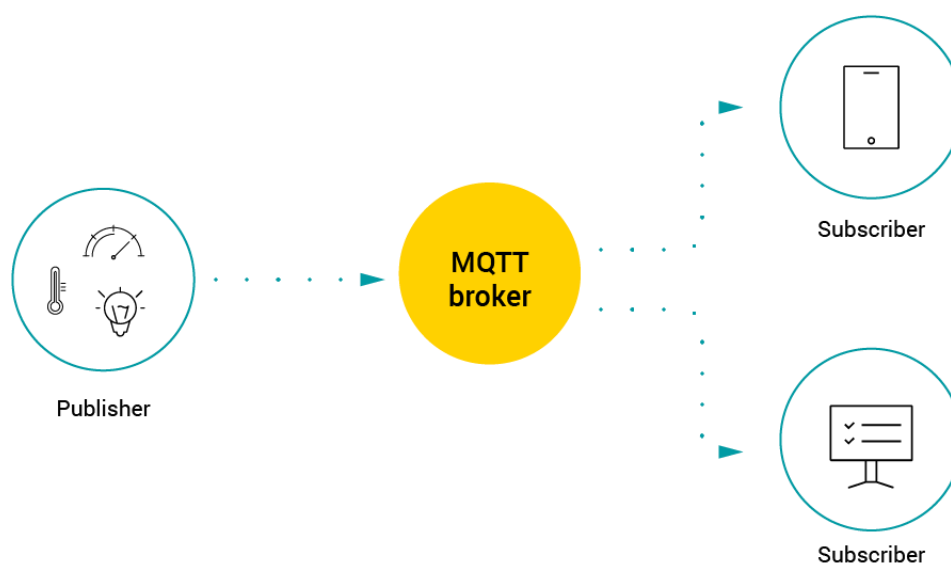


Рис. 1.7. MQTT

Незважаючи на свої характеристики, MQTT може бути проблематичним для деяких дуже обмежувальних пристроїв через факт передачі повідомлень через TCP і керування довгими іменами тем. Це вирішується за допомогою варіанту MQTT-SN, який використовує UDP та підтримує індексацію назв теми. Однак, незважаючи на широке поширення, MQTT не підтримує чітко визначену модель представлення даних і структури керування пристроями, що робить реалізацію його можливостей керування даними та керування пристроями повністю залежною від платформи або постачальника.

Wi-Fi. Для створення мережі Wi-Fi потрібні пристрої, які можуть надсилати бездротові сигнали, наприклад такі пристрої, як телефони, комп'ютери чи маршрутизатори. У домашніх умовах маршрутизатор використовується для передачі інтернет-з'єднання з загальнодоступної мережі в приватну домашню або офісну мережу. Wi-Fi забезпечує підключення до Інтернету для пристроїв поблизу, які знаходяться в певному діапазоні. Інший спосіб використання Wi-Fi — це створення точки доступу Wi-Fi, тобто телефони або комп'ютери можуть

використовувати бездротове або дротове підключення до Інтернету з іншими пристроями, передаючи сигнал.

Wi-Fi використовує радіохвилі, які передають інформацію на певних частотах, наприклад на каналах 2,4 ГГц або 5 ГГц. Обидва діапазони частот мають ряд каналів, через які можуть працювати різні бездротові пристрої, що допомагає розподілити навантаження так, щоб окремі з'єднання пристроїв не переривались. Це значною мірою запобігає переповненню бездротових мереж.

Радіус дії 100 метрів є типовим діапазоном стандартного Wi-Fi-з'єднання. Найбільш поширений діапазон, однак, обмежений 10-35 метрами. На ефективне покриття мережі значною мірою впливає потужність антени або частота передачі. Дальність і швидкість підключення до Інтернету через Wi-Fi залежить від середовища та від того, чи забезпечує воно внутрішнє чи зовнішнє покриття. Таким чином, швидкість роботи різних пристроїв, які використовують інтернет-з'єднання Wi-Fi, збільшується в міру наближення комп'ютера до основного джерела, а швидкість зменшується, коли комп'ютер віддаляється від джерела.

ZigBee. Мережі на базі ZigBee характеризуються низьким споживанням енергії, низькою пропускну здатністю (до 250 кбіт/с) і діапазоном підключення до 100 метрів між вузлами. Типові застосування включають сенсорні мережі, персональні мережі (WPAN), домашню автоматизацію, системи сигналізації та системи моніторингу.

Його початкова специфікація була визнана стандартом IEEE в 2003 році, а перші сумісні з нею OEM модулі ZigBee з'явилися в масових продажах на початку 2006 року.

ZigBee був розроблений як стандарт для самонастроюваних радіомереж малого радіусу дії, призначений для використання в телеметричних системах, для зв'язку між різними типами датчиків, моніторингових пристроїв, а також для бездротового зчитування результатів вимірювань з лічильників енергії та тепла, і т. д. Стандарт ZigBee - це відносно простий, стійкий до помилок зв'язку та

несанкціонованого зчитування протокол обміну пакетними даними, який часто реалізується в пристроях з відносно невеликими вимогами, наприклад, мікроконтролери, датчики тощо.

ZigBee простий в установці та обслуговуванні, оскільки він заснований на топології самозбирання та самовідновлення сітки. Він також легко масштабується до тисяч вузлів, і сьогодні є багато постачальників, які пропонують пристрої, які підтримують цей відкритий стандарт.

Bluetooth – це технологія, яка дозволяє бездротове підключення різних електронних пристроїв, таких як телефон, клавіатура, комп'ютер, ноутбук, миша, КПК, принтер, гарнітура або гучномовець тощо. Якщо ви хочете отримати більш схоже на вікі визначення, це відкритий стандарт, описаний у специфікації IEEE 802.15.1, і його технічна специфікація включає три класи потужності передачі ERP 1-3 з діапазоном, відповідно, 100, 10 і 1 метр на відкритому просторі. Найпоширенішим класом є другий (10м), який дозволяє підключати пристрої, що знаходяться в різних кімнатах і навіть на різних поверхах.

Стандарт використовує радіохвилі в діапазоні частот ISM 2,4 ГГц, а пристроєм, що дозволяє використовувати цей стандарт, є адаптер Bluetooth. У технології Bluetooth дані надсилаються у вигляді пакетів на один із 79 каналів (у випадку найстарішого стандарту Bluetooth 1.0) із пропускнуою здатністю 1 МГц, що забезпечує максимальну швидкість передачі 721 кбіт/с. У разі останнього стандарту Bluetooth 4.0 є 40 каналів із пропускнуою здатністю 2 МГц, що гарантує максимальну передачу даних до 3 Мб/с. Варто знати, що новіші стандарти Bluetooth, які гарантують швидшу передачу даних і більшу безпеку, також сумісні зі старими версіями.

XMPP (Extensible Messaging and Presence Protocol - Розширюваний протокол обміну повідомленнями та присутності) - Розроблений у 1999 році спільнотою відкритого коду Jabber і спочатку призначений для обміну повідомленнями в реальному часі, цей протокол зв'язку IoT для проміжного

програмного забезпечення, орієнтованого на повідомлення, заснований на мові XML. Це дозволяє в реальному часі обмінюватися структурованими, але розширюваними даними між двома або більше мережевими клієнтами.

З моменту свого створення XMPP широко застосовувався як протокол зв'язку. З часом і з появою легкої специфікації XMPP: XMPP-IoT, вона стала використовуватися в контексті Інтернету речей. Будучи відкритим стандартом, який підтримується спільнотою, сильними сторонами XMPP IoT є можливості адресації та масштабованості, що робить його ідеальним для розгортання Інтернету речей, орієнтованого на споживача.

Серед недоліків використання XMPP в IoT-комунікації слід зазначити, що він не пропонує ні якості обслуговування, ні наскрізного шифрування. Через ці обмеження, серед іншого, передбачається, що його застосування в IoT залишиться слабко пов'язаним з галуззю, оскільки протокол точно не стане стандартом, що використовується день у день для цілей обміну даними та керування пристрої з обмеженими ресурсами, як і MQTT або LwM2M.

DDS (Data-Distribution Service - Служба розповсюдження даних) - Протокол DDS розроблено на основі методології публікації-підписки. Розроблений Групою управління об'єктами (OMG), протокол DDS для зв'язку M2M в реальному часі забезпечує масштабований, надійний, високопродуктивний і сумісний обмін даними між підключеними пристроями незалежно від апаратного забезпечення та програмної платформи. DDS підтримує архітектуру без брокерів і багатоадресну передачу для забезпечення високої якості QoS і забезпечення взаємодії. Архітектура протоколу DDS базується на рівні публікації-підписки, орієнтованого на дані (DCPS) і додатковому рівні локальної реконструкції даних (DLRL). У той час як рівень DCPS відповідає за обізнаний з ресурсами, масштабований та ефективний розподіл даних серед абонентів, DLRL пропонує інтерфейс для функціональних можливостей DCPS, що дозволяє передавати дані між об'єктами, підключеними до IoT.

Незважаючи на те, що DDS не є типовим рішенням IoT, він все ще знаходить своє застосування в деяких промислових розгортаннях Інтернету речей, таких як: контроль повітряного руху, управління інтелектуальними мережами, автономні транспортні засоби, транспортні системи, робототехніка, виробництво електроенергії та медичні послуги. Загалом, DDS можна використовувати для керування обміном даними між легкими пристроями та взаємоз'єднанням великих високопродуктивних сенсорних мереж. Він також може надсилати та отримувати дані з хмари.

AMQP (Advanced Message Queuing Protocol - Розширений протокол черги повідомлень) - це відкритий стандартний протокол типу публікації/підписки, який виник у 2003 році, що сягає корінням у сектор фінансових послуг. Незважаючи на те, що він набув популярності в інформаційно-комунікаційних технологій, його використання все ще досить обмежене в індустрії Інтернету речей. Специфікація AMQP описує такі функції, як орієнтація повідомлень, черга, маршрутизація (включаючи "точка-точка" та "опублікування та підписки", надійність і безпеку). Ймовірно, найбільшою перевагою AMQP є його надійна комунікаційна модель. AMQP може гарантувати повні транзакції, що, хоча й корисно, не завжди вимагає додатків Інтернету речей.

Через свою важкість AMQP не підходить для сенсорних пристроїв з обмеженою пам'яттю, потужністю або пропускнуою здатністю мережі, але для окремих випадків використання IoT це може бути єдиним протоколом, життєздатним для наскрізного застосування, включаючи такі приклади, як промислове важке обладнання або системи SCADA, де пристрої та мережа, як правило, значно більші.

LwM2M (Lightweight M2M - Легкий M2M) - Що відрізняє LwM2M від інших протоколів, що застосовуються в Інтернеті речей, так це те, що він був спеціально розроблений, щоб задовольнити вимоги до всебічної роботи з пристроями з обмеженими ресурсами. Запущений у 2014 році Open Mobile Alliance

(зараз OMA SpecWorks), він забезпечує чітко визначений стандарт для передачі даних IoT та керування пристроями.

HTTP (Hypertext Transfer Protocol - протокол передачі гіпертексту) - це набір правил для передачі файлів, таких як текст, зображення, звук, відео та інші мультимедійні файли, через Інтернет. Як тільки користувач відкриває свій веб-браузер, він опосередковано використовує HTTP. HTTP — це прикладний протокол, який працює поверх набору протоколів TCP/IP, що становить основу Інтернету. Остання версія HTTP — HTTP/2, яка була опублікована в травні 2015 року. Це альтернатива своєму попереднику, HTTP 1.1, але не застаріває. Через протокол HTTP ресурси обмінюються між клієнтськими пристроями та серверами через Інтернет. Клієнтські пристрої надсилають серверам запити на ресурси, необхідні для завантаження веб-сторінки; сервери надсилають відповіді назад клієнту для виконання запитів. Запити та відповіді спільно використовують піддокументи, такі як дані про зображення, текст, макети тексту тощо, які об'єднуються клієнтським веб-браузером для відображення повного файлу веб-сторінки.

На додаток до файлів веб-сторінок, які він може обслуговувати, веб-сервер містить демон HTTP, програму, яка чекає на запити HTTP і обробляє їх, коли вони надходять. Веб-браузер — це HTTP-клієнт, який надсилає запити на сервери. Коли користувач браузера вводить запити файлів, «відкриваючи» веб-файл, вводячи URL-адресу або натискаючи гіпертекстове посилання, браузер створює HTTP-запит і надсилає його на адресу Інтернет-протоколу (IP-адресу), зазначену в URL-адресі. Демон HTTP на сервері призначення отримує запит і надсилає назад запитаний файл або файли, пов'язані із запитом. Щоб розширити цей приклад, користувач хоче відвідати TechTarget.com. Користувач вводить веб-адресу, і комп'ютер надсилає запит «GET» на сервер, на якому розміщена ця адреса. Цей запит GET надсилається за допомогою HTTP і повідомляє серверу TechTarget, що користувач шукає код HTML (мова гіпертекстової розмітки), який

використовується для структурування та надання вигляду сторінки входу. Текст цієї сторінки входу включається у відповідь HTML, але інші частини сторінки, зокрема її зображення та відео, запитуються окремими HTTP-запитами та відповідями. Чим більше запитів зроблено — наприклад, для виклику сторінки, яка містить багато зображень — тим більше часу знадобиться серверу, щоб відповісти на ці запити і системі користувача завантажити сторінку.

Коли ці пари запит/відповідь надсилаються, вони використовують TCP/IP для скорочення та транспортування інформації в невеликих пакетах двійкових послідовностей одиниць і нулів. Ці пакети фізично надсилаються через електричні дроти, волоконно-оптичні кабелі та бездротові мережі.

Запити та відповіді, які сервери та клієнти використовують для обміну даними один з одним, складаються з коду ASCII. Запити вказують, яку інформацію клієнт шукає від сервера; відповіді містять код, який клієнтський браузер переведе на веб-сторінку.

Представлені технології є дуже перспективними та цікавими, кожна з них має свої переваги, але через те, що система буде мати живлення від адаптеру елетромережі, а як мережу використовувати Wi-Fi, то у зниженні електроспоживання чи розміру пакету даних немає сенсу. Серед усіх протоколів було обрано HTTP (HTTPS), так як для нього можна знайти найбільше інформації та документацій, і у цілому він є простішим у налаштуванні, у порівнянні з іншими протоколами. Також однією із причин вибору саме HTTP було те, що він не потребує додаткового обладнання чи серверу, як MQTT, якому потрібен брокер пакетів.

1.6 Моніторинг

Основним призначенням даної системи є моніторинг показників системи очищення води у реальному часі. Тобто через певний встановлений час оновлення на веб-додатку будуть відображатись наступні дані:

- 1) ідентифікатор системи (порядковий номер);
- 2) швидкість потоку води;
- 3) температура води;
- 4) тиск у водопостачанні;
- 5) TDS неочищеної води (кількість розчинених солей);
- 6) TDS очищеної води (кількість розчинених солей);
- 7) наявність протіку води з фільтру;
- 8) час і дата надходження даних;
- 9) залишок ресурсу у літрах;
- 10) залишок ресурсу у днях.

Але постає питання – звідки взяти ресурс картриджів у літрах? Щоб відповісти на це питання, треба зрозуміти що це таке. Ресурс картриджу – це такий проміжок часу (або кількість літрів) після якого вода починає фільтруватись недостатньо. Зазвичай різні виробники позначають різну кількість літрів/днів, і насправді даний показник є індивідуальним, у кожного споживача цей ліміт буде унікальним, так як буде залежати від його параметрів системи – насільки чиста вхідна вода, яка температура (так як вона також впливає на розмноження бактерій у картриджах, тощо). Також не слід забувати, що показники від виробника можуть бути спекулятивно занижені, тобто з метою підвищення продаж товару вони не будуть відповідати дійсності.

Вирішити дану проблему можна за допомогою даної системи. Якщо купити потрібні картриджі, вставити їх, і запустити систему фільтрації води, з плином часу можна буде спостерігати за зміною вихідного показника TDS, що буде свідчити про погіршення роботи фільтру. Для цього також потрібно буде визначити вищий поріг чистоти води. (рис.1.8).

Згідно з нормами України, а саме ДСТУ 7525:2014, концентрація домішки у воді повинна бути не більше 7 мг-екв/л, що приблизно дорівнює 350 ppm. Але, якщо у воді TDS більше ніж 170, вона буде неприємною для вживання, чи як її просто

називають – жорсткою. Через таку воду організм людини буде отримувати непотрібні йому домішки, а побутова техніка – накип. Після того, як було встановлено потрібний параметр для моніторингу, потрібно його відстежити. Але на це потрібно до півроку часу! І тут на допомогу приходить вирішення даної проблеми – прогнозування.

Висновки до розділу 1

У першому розділі роботи було розглянуто компоненти і модулі, які повинні стати основою майбутньої системи. Було розглянуто різноманітні типи і види фільтрів для води, їх призначення і особливості. Для створення системи було обрано фільтр типу зворотній осмос. У якості мікроконтролера було обрано Arduino ESP32, так як це єдина модель, яка задовольняє потреби майбутньої системи по своїм показникам.

Також було розглянуто протоколи, види і типи зв'язку, мови програмування, і було прийняте рішення створювати систему на мові Arduino, з відправленням даних через бездротову мережу Wi-Fi, за допомогою протоколу http, методом POST.

Додатково було обрано основні показники, які буде бачити споживач, а саме - швидкість потоку води, температура води, тиск у водопостачанні, TDS неочищеної води (кількість розчинених солей), TDS очищеної води (кількість розчинених солей), наявність протіку води з фільтру, час і дата надходження даних, залишок ресурсу у літрах, залишок ресурсу у днях.

2 ПРОГНОЗУВАННЯ ПАРАМЕТРІВ

2.1 Що таке прогнозування, його моделі та види

Прогнозування – це процес передбачення майбутнього стану об'єкта або явища на основі аналізу його минулого і сьогодення, систематичної оцінки інформації про якісні та кількісні характеристики майбутнього розвитку вибраного об'єкта або явища. Результатом прогнозування є прогноз - розуміння майбутнього і ймовірного розвитку поточних тенденцій для конкретного феноменологічного об'єкта в майбутньому. Інструменти прогнозування працюють на основі низки різних моделей і алгоритмів, які можна застосувати до широкого кола випадків використання. Визначення того, який метод прогнозного моделювання є найкращим, є ключем до отримання максимальної віддачі від рішень прогнозної аналітики та використання даних для глибокого прийняття рішень. Які моделі прогнозної аналітики є найбільш поширеними? Який алгоритм прогнозування буде найбільш корисним для їх годування?

П'ять найкращих моделей прогнозної аналітики:

1) модель класифікації. У певному сенсі модель класифікації є найпростішою з кількох моделей прогнозної аналітики, які будуть розглянуті. Вона поділяє дані на категорії на основі того, що було визначено з історичних даних. Моделі класифікації найкраще відповідають на запитання так чи ні, забезпечуючи широкий аналіз, який допомагає керувати рішучими діями. Ці моделі можуть відповісти на наступні питання:

- для роздрібних продавців: "Цьому клієнту буде відмовлено?" Для постачальників позики: "Чи буде схвалений цей кредит?" або "Чи, ймовірно, цей заявник не виконує зобов'язання?";

- для постачальників послуг онлайн-банкінгу: "Це шахрайська операція?"
Широта моделі класифікації — і те, наскільки легко її можна перенавчати за

допомогою нових даних — означає, що її можна застосувати до багатьох різних галузей;

2) модель кластеризації. Моделі кластеризації класифікують дані в окремі вкладені розумні групи на основі подібних атрибутів. Якщо компанія, що займається електронною реалізацією взуття, хотіла б розпочати цільову маркетингову кампанію для своїх клієнтів, вона могла б переглянути сотні тисяч записів і створити індивідуальну стратегію для кожного з них. Але чи це найефективніше використання часу? Можливо ні. Використовуючи моделі кластеризації, вони можуть швидко сегментувати клієнтів на подібні групи на основі загальних характеристик і розробляти стратегії для кожної групи у більшому масштабі. Інші види використання цього методу прогнозного моделювання можуть включати групування заявників у «розумні комірки» на основі атрибутів позики, виявлення районів високого рівня злочинності в містах та проведення порівняльного аналізу даних клієнтів для визначення глобальних моделей використання;

3) модель прогнозу. Однією з найбільш широко використовуваних моделей у прогнозній аналітиці є прогнозна модель, яка прогнозує значення метрики шляхом оцінки цінності нових даних на основі знання історичних даних. Модель можна застосувати будь-де, де є історичні числові дані. Сценарії включають:

- модель може оцінити, скільки клієнтів вони, імовірно, буде обслуговано протягом певного тижня;
- для колл-центрів можна передбачити, скільки дзвінків служби підтримки вони отримають за годину;
- взуттєві магазини можуть розрахувати, скільки запасів їм потрібно мати, щоб задовольнити попит під час розпродажу.

Прогнозна модель також враховує кілька вхідних параметрів. Якщо власник ресторану хоче передбачити кількість клієнтів, які вона, швидше за все, обслуговуватиме наступного тижня, модель враховуватиме фактори, які можуть

вплинути на це, наприклад: чи є поблизу якась подія? Який прогноз погоди? Чи зникне хвороба?;

4) модель викидів. Модель викидів орієнтована на аномальні записи даних у наборі даних. Він може ідентифікувати аномальні фігури як самі по собі, так і в поєднанні з іншими числами та категоріями. Зафіксовано сплеск дзвінків у службу підтримки, що може свідчити про збій продукту, що може призвести до відкликання. Пошук аномальних даних у транзакціях або в страхових виплатах для виявлення шахрайства.

- можна знаходити незвичну інформацію у журналах і ознаки незапланованого простоя;

- модель викидів особливо корисна для прогнозувальної аналітики в роздрібній торгівлі та фінансах. Наприклад, при виявленні шахрайських трансакцій модель може оцінити не тільки суму, а й місце, час, історію покупок і характер покупки (тобто покупка електроніки на 1000 доларів США не так імовірна буде шахрайською, як покупка такої ж суми на книги чи комунальні послуги).

5) модель часового ряду. Модель часового ряду складається з фіксованого ряду точок даних, що використовують час як вхідний параметр. Він використовує дані минулого року для розробки числового показника і використовує цей показник для прогнозування даних на наступні три-шість тижнів. Приклади цієї моделі включають кількість дзвінків, отриманих за день за останні три місяці, продажі за останні 20 кварталів або кількість пацієнтів, госпіталізованих до певної лікарні за останні шість тижнів. Це потужний інструмент для розуміння того, як одиничні показники змінюються з часом з більшою точністю, ніж прості середні. Він також враховує сезон року або події, які можуть вплинути на показник. Якщо власник салону хоче передбачити, скільки людей, ймовірно, відвідають його магазин, він може використовувати приблизний метод для усереднення загальної кількості відвідувачів за останні 90 днів. Однак зростання не завжди є статичним або лінійним, і моделі часових рядів можуть краще імітувати експоненційне зростання

і краще узгоджувати модель з тенденціями компанії. Вона також може прогнозувати декілька проектів або регіонів одночасно, а не лише по одному.

Отже, визначивши призначення та основні проблеми для застосування моделей прогнозування було вирішено використовувати модель часового ряду, так як при поступовій витраті ресурсу зростання TDS буде експоненційним, і ця модель підійде найкраще.

2.2 Прогнозування часових рядів

Прогнозування часових рядів відбувається, коли ви робите наукові прогнози на основі даних із історичними мітками часу. Це передбачає створення моделей за допомогою історичного аналізу та використання їх для спостереження та прийняття майбутніх стратегічних рішень. Важлива відмінність у прогнозуванні полягає в тому, що на момент роботи майбутні результати повністю недоступні і можуть бути оцінені лише шляхом ретельного аналізу та підтверджених доказів.

Прогнозування часових рядів — це процес аналізу даних часових рядів із використанням статистики та моделювання для прогнозування та прийняття стратегічних рішень. Це не завжди точний прогноз, і прогнозовані ймовірності можуть значно відрізнятися, особливо якщо мова йде про змінні, які часто коливаються в даних часових рядів, і факторах, які знаходяться поза нашим контролем. Як правило, чим повніші ми маємо дані, тим точнішими будуть прогнози. Хоча прогноз і «передбачення» часто означають одне й те саме, є чітка різниця. У деяких сферах прогнози можуть бути пов'язані з даними в певний момент у майбутньому, тоді як передбачення часто пов'язані з майбутніми даними. Прогнозування послідовності часто використовується в поєднанні з аналізом часових рядів. Аналіз часових рядів включає розробку моделей для розуміння даних, щоб зрозуміти першопричину. Прогнозування має широкий спектр застосування в різних галузях промисловості. Воно має багато практичних застосувань, зокрема: прогнозування погоди, прогнозування клімату, економічне прогнозування, прогнозування здоров'я, інженерне прогнозування, фінансове

прогнозування, прогнозування роздрібної торгівлі, бізнес-прогнозування, прогнозування екологічних досліджень, прогнозування соціальних досліджень тощо. По суті, будь-хто, хто має послідовні історичні дані, може проаналізувати ці дані, використовуючи методи аналізу часових рядів, а потім моделювати, та прогнозувати. Для деяких галузей цілісний аналіз часових рядів призначений для спрощення прогнозування. Деякі методи, такі як розширена аналітика, можуть навіть автоматично вибирати прогнози з інших статистичних алгоритмів, якщо вони забезпечують впевненість того, що може статися в майбутньому.

Перше, на що потрібно звернути увагу, це кількість доступних даних – чим більше у вас точок у часі - тим краще буде результат. Це константа для всіх типів аналізу, і аналіз часових рядів прогнозування не є винятком. Проте прогнозування значною мірою залежить від кількості даних, можливо, навіть більше, ніж інші аналізи. Він базується безпосередньо на минулих і поточних даних. Чим більше даних потрібно екстраполювати, тим менш точними будуть ваші прогнози.

Часові горизонти. Важливе значення має і термін часовий горизонт. Це - фіксований момент часу, на якому закінчується процес (наприклад, прогноз). Набагато легше передбачити коротші часові горизонти з меншою кількістю змінних, ніж більш тривалі часові горизонти. Чим далі ви йдете, тим менш передбачувані змінні. Якщо вам не вистачає довгострокових записаних даних, але у вас є багато короткострокових даних, ви можете створити короткострокові прогнози.

Динамічні та статичні стани. Стан вашого прогнозу та даних впливає на те, коли ви їх використовуєте. Чи є прогнози динамічними чи статичними? Якщо прогноз статичний, він фіксується після його створення, тому переконайтеся, що ваших даних достатньо, щоб зробити передбачення. Однак динамічні прогнози можна постійно оновлювати в міру надходження нової інформації. Це означає, що ви можете отримувати менше даних під час прогнозування, а потім отримувати точніші прогнози під час додавання даних.

Якість даних. Під час аналізу найкращий аналіз потрібно розуміти, що дані повинні буди достовірними та правдивими. Брудні, неправильно оброблені, надмірно оброблені або неправильно зібрані дані можуть серйозно спотворити результати і спричинити вкрай неточні прогнози. Ось деякі загальні рекомендації щодо якості даних:

- 1) переконайтеся, що дані повні;
- 2) дані не повинні повторюватись;
- 3) дані зібрані вчасно та послідовно;
- 4) дані мають стандартний і дійсний формат.

Коли маєте справу з аналізом часових рядів, важливо регулярно збирати дані протягом періоду часу, який відстежується. Це допомагає врахувати тенденції даних, циклічність поведінки та сезонність. Це також може допомогти визначити, чи є викиди справді екстремальними або є частиною більшого циклу. Прогалини в даних можуть маскувати циклічні або сезонні коливання, що призводить до спотворених прогнозів.

2.3 Моделі прогнозування часових рядів

Буде розглянуто наступні моделі прогнозування часових рядів:

- 1) Naïve, Snaïve;
- 2) сезонна декомпозиція (+ будь-яка модель);
- 3) експоненційне згладжування;
- 4) ARIMA, SARIMA;
- 5) GARCH;
- 6) динамічні лінійні моделі;
- 7) TBATS;
- 8) модель «Пророк»;
- 9) NNETAR;
- 10) LSTM.

Метою є прогнозування індексу чистоти води. Середня абсолютна помилка (MAE) буде використовуватись для оцінки продуктивності моделей.

2.3.1 Naïve, Snaïve

У наївній моделі прогнози для кожного горизонту відповідають останньому спостережуваному значенню.

$$Y\left(t + \frac{h}{t}\right) = Y(t) \quad (2.1)$$

Цей вид прогнозу передбачає, що стохастична модель, яка генерує часові ряди, є випадковим блуканням. Розширення наївної моделі дає модель SNaïve (Seasonal Naïve). Припускаючи, що часовий ряд має сезонну складову і що період сезонності дорівнює T , прогнози, надані моделлю SNaïve, отримують вид:

$$Y\left(t + \frac{h}{t}\right) = Y(t + h - T) \quad (2.2)$$

Тому прогнози для наступних T кроків часу дорівнюють попереднім T крокам часу. У додатку прогноз SNaïve на наступний рік дорівнює минулорічним спостереженням (Рис.2.1). Ці моделі часто використовуються як еталонні моделі. Наступні графіки показують прогнози, отримані з двома моделями на 2007 рік. (рис. 2.1)

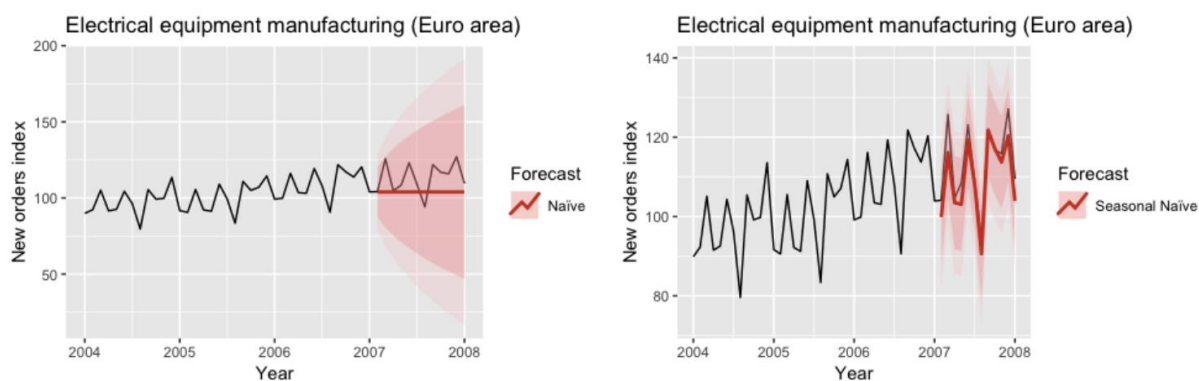


Рис. 2.1. Моделі Naïve та SNaïve

2.3.2 Сезонна декомпозиція (+ будь-яка модель)

Якщо дані показують певну сезонність (наприклад, щодня, щотижня, квартал, рік), може бути корисно розкласти вихідний часовий ряд на суму трьох компонентів:

$$Y(t) = S(t) + T(t) + R(t) \quad (2.3)$$

де $S(t)$ – сезонний компонент, $T(t)$ – компонент трендового циклу, а $R(t)$ – компонент залишку. Існує кілька методів оцінки такого розкладання. Найбільш основний з них називається класичним розкладом. Класична декомпозиція була розширена кількома способами. Її розширення дозволяють:

- 1) мати непостійну сезонність;
- 2) обчислювати початкове та останнє значення розкладання;
- 3) уникати надмірного розгладжування.

Щоб отримати огляд методів декомпозиції часових рядів, натисніть тут. Ми скористаємося перевагами розкладання STL, яке, як відомо, є універсальним і надійним.

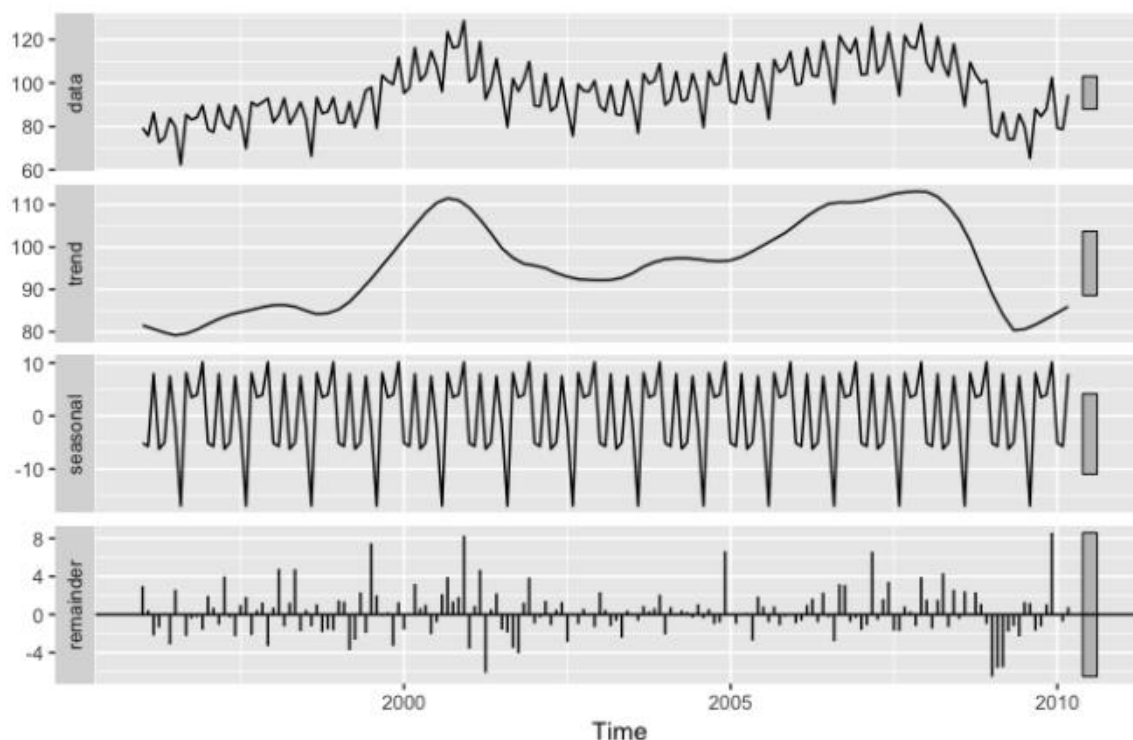


Рис. 2.2. Розклад STL за даними індексу промислового виробництва

Один із способів використання декомпозиції для цілей прогнозування:

- 1) розкладіть часовий ряд навчання за допомогою певного алгоритму декомпозиції (наприклад, STL (Рис. 2.2)): $Y(t) = S(t) + T(t) + R(t)$;
- 2) обчисліть сезонно скоригований часовий ряд $Y(t) - S(t)$. Використовуйте будь-яку модель, яка вам подобається, щоб спрогнозувати еволюцію сезонно скоригованого часового ряду;
- 3) додайте до прогнозів сезонність останнього періоду часу в часовому ряді (у нашому випадку, встановлений $S(t)$ за минулий рік).

На малюнку показано сезонно скоригований часовий ряд індексу промислового виробництва (рис. 2.3).



Рис. 2.3. Сезонно скоригований часовий ряд індексу промислового виробництва

Наступний графік (Рис.2.4) показує прогнози, отримані на 2007 рік за допомогою розкладання STL та наївної моделі, щоб відповідати сезонно скоригованим часовим рядам.

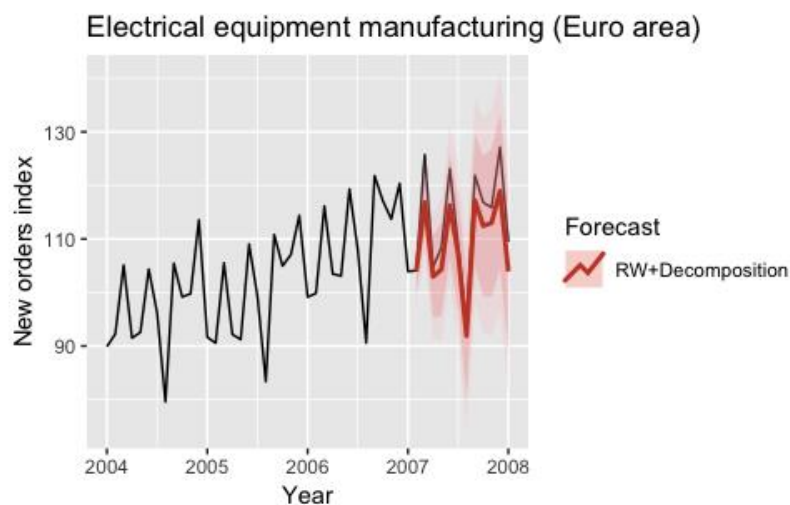


Рис. 2.4. Прогнози, отримані за допомогою розкладання STL та наївної моделі

2.3.3 Експоненційне згладжування

Експоненціальне згладжування є одним з найуспішніших класичних методів прогнозування. У своїй основній формі це називається простим експоненційним згладжуванням, а його прогнози подаються так:

$$Y\left(t + \frac{h}{t}\right) = \alpha y(t) + \alpha(1 - \alpha)y(t - 1) + \alpha(1 - \alpha)^2 y(t - 2) + \dots \quad (2.4)$$

де $0 < \alpha < 1$.

Прогнози дорівнюють середньозваженому показнику минулих спостережень, а відповідні вагові коефіцієнти зменшуються експоненціально, коли ми повертаємося в минуле. Було запропоновано кілька розширень простого експоненційного згладжування, щоб включити тенденцію або затухання тенденції та сезонність. Графіки (рис.2.5) показують прогнози, отримані на 2007 рік за допомогою моделей експоненційного згладжування (автоматично вибраних), щоб відповідати як оригінальному, так і сезонно скоригованим часовим рядам.

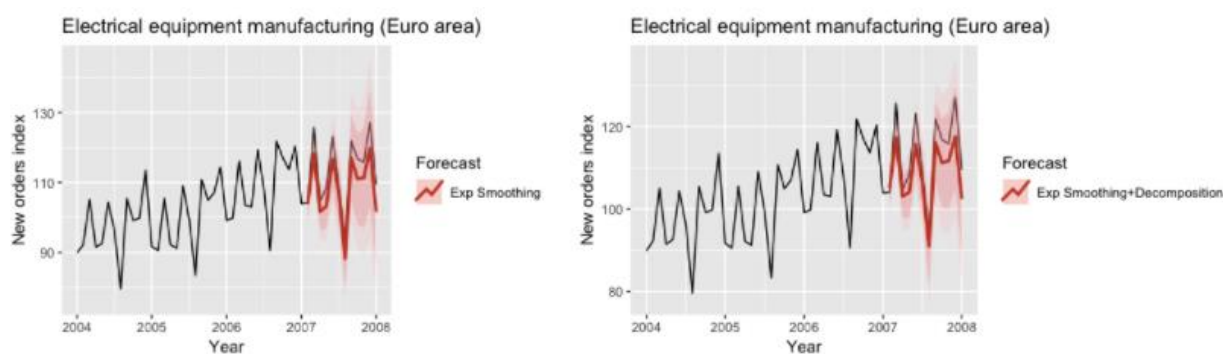


Рис. 2.5. Прогнози, отримані за допомогою моделей експоненційного згладжування

2.3.4 ARIMA, SARIMA

Що стосується експоненційного згладжування, то моделі ARIMA також є одними з найбільш широко використовуваних підходів для прогнозування часових рядів. Назва є акронімом від AutoRegressive Integrated Moving Average.

У моделі авторегресії прогнози відповідають лінійній комбінації минулих значень змінної. У моделі ковзного середнього прогнози відповідають лінійній комбінації попередніх помилок прогнозу. В основному, моделі ARIMA поєднують ці два підходи. Оскільки вони вимагають, щоб часові ряди були стаціонарними, диференціювання (інтегрування) часового ряду може бути необхідним кроком, тобто розгляд часового ряду відмінностей замість вихідного.

Модель SARIMA (Seasonal ARIMA) розширює ARIMA, додаючи лінійну комбінацію сезонних минулих значень та/або помилок прогнозу. Наступні графіки (Рис. 2.6) показують прогнози, отримані на 2007 рік з використанням моделі SARIMA та моделі ARIMA для сезонно скориговного часового ряду.

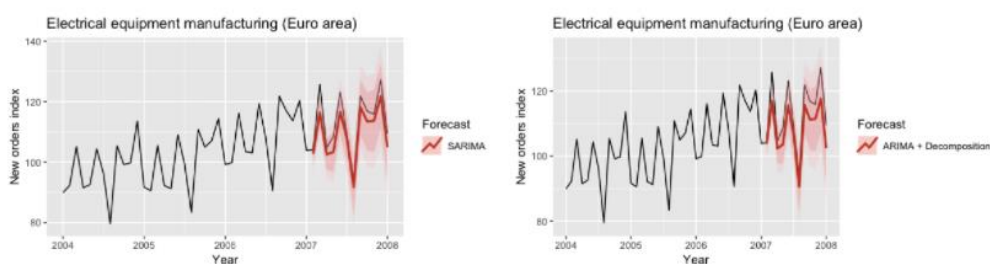


Рис. 2.6. Прогнози, отримані з використанням моделі SARIMA та моделі ARIMA

2.3.5 GARCH

Попередні моделі передбачали, що умови помилки в випадкових процесах, які породжують часові ряди, були гомоскедастичними, тобто з постійною дисперсією. Натомість модель GARCH припускає, що дисперсія умов помилки слідує за процесом авторегресивного ковзного середнього (ARMA), отже, дозволяє йому змінюватися в часі. Це особливо корисно для моделювання фінансових часових рядів, волатильність яких змінюється в часі. Назва є акронімом від Generalized Autoregressive Conditional Heteroskedasticity. Наступні графіки (Рис.2.7) показують прогнози, отримані на 2007 рік за допомогою моделі GARCH, щоб відповідати сезонно скоригованим часовим рядам.

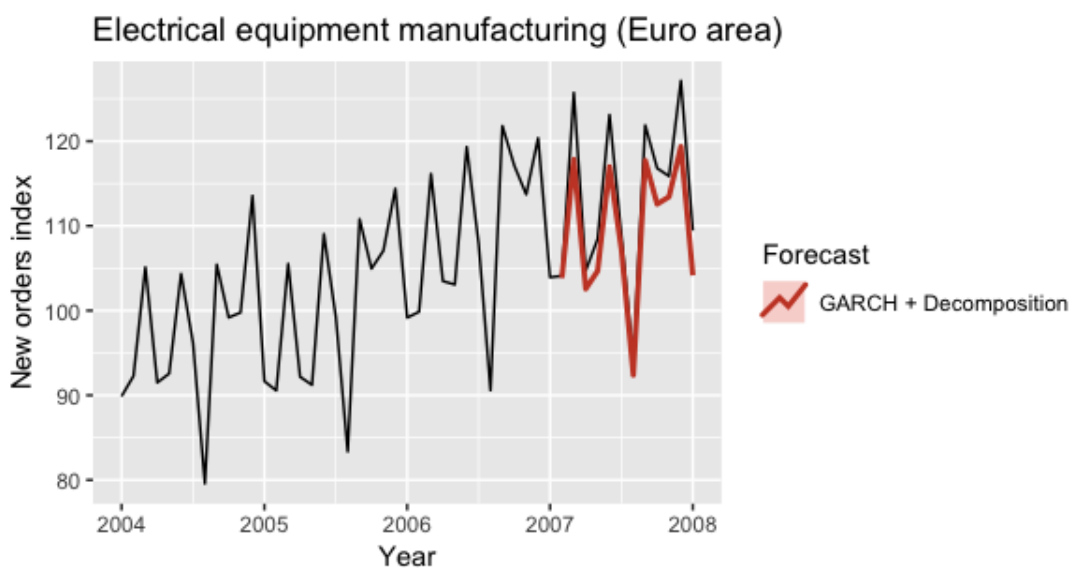


Рис. 2.7. Прогнози, отримані за допомогою моделі GARCH

2.3.6 Динамічні лінійні моделі

Динамічні лінійні моделі представляють інший клас моделей для прогнозування часових рядів. Ідея полягає в тому, що в кожен момент часу t ці моделі відповідають лінійній моделі, але коефіцієнти регресії змінюються з часом. Нижче наведено приклад динамічної лінійної моделі.

$$y(t) = \alpha(t) + t\beta(t) + w(t) \quad (2.5)$$

$$\alpha(t) = \alpha(t - 1) + m(t) \quad (2.6)$$

$$\beta(t) = \beta(t - 1) + r(t) \quad (2.7)$$

$$w(t) \sim N(0, W), m(t) \sim N(0, M), r(t) \sim N(0, R) \quad (2.8)$$

У попередній моделі коефіцієнти $\alpha(t)$ і $\beta(t)$ слідують за процесом випадкового блукання. Динамічні лінійні моделі можна природним чином моделювати в байєсівській структурі, однак методи оцінки максимальної ймовірності все ще доступні.

Наступний графік (Рис.2.8) показує прогнози, отримані на 2007 рік за допомогою динамічної лінійної моделі, щоб відповідати сезонно скоригованим часовим рядам. Через великі обчислювальні витрати мені довелося зберегти модель надзвичайно простою, що призвело до поганих прогнозів.

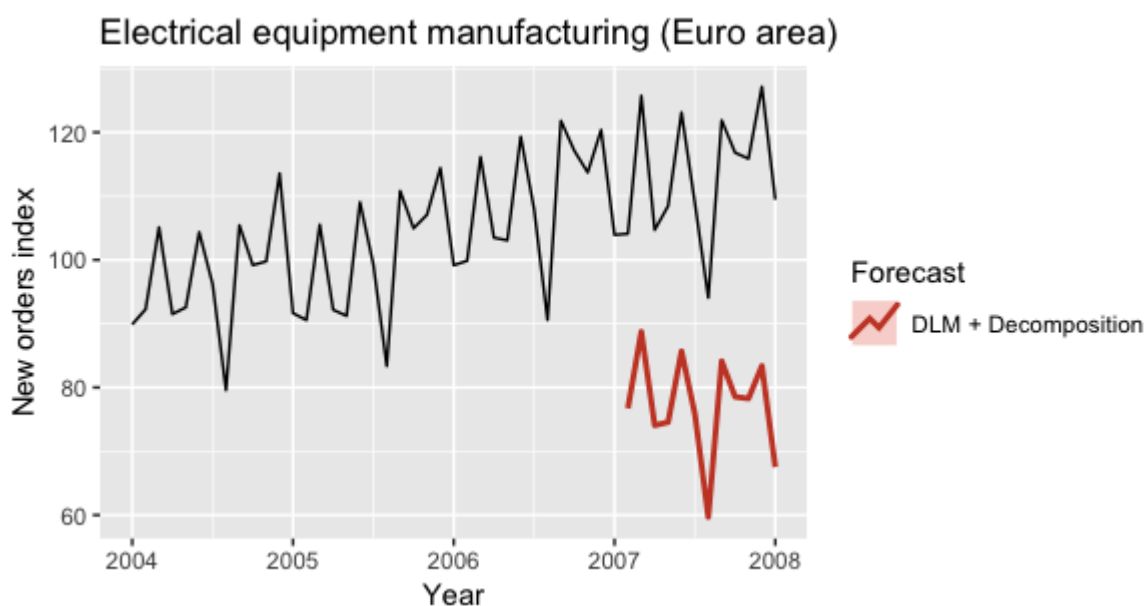


Рис. 2.8. Прогнози, отримані за допомогою динамічної лінійної моделі

2.3.7 TBATS

Модель TBATS — це модель прогнозування, заснована на експоненційному згладжуванні. Назва є акронімом для тригонометричних, перетворення Бокса-Кокса, помилок ARMA, тенденцій та сезонних компонентів.

Головною особливістю моделі TBATS є її здатність працювати з кількома сезонностями шляхом моделювання кожної сезонності за допомогою тригонометричного представлення на основі рядів Фур'є. Класичним прикладом комплексної сезонності є щоденні спостереження за обсягами продажів, які часто мають як тижневу, так і річну сезонність. Наступний графік (Рис.2.9) показує прогнози, отримані на 2007 рік за допомогою моделі TBATS для відповідності часовим рядам.

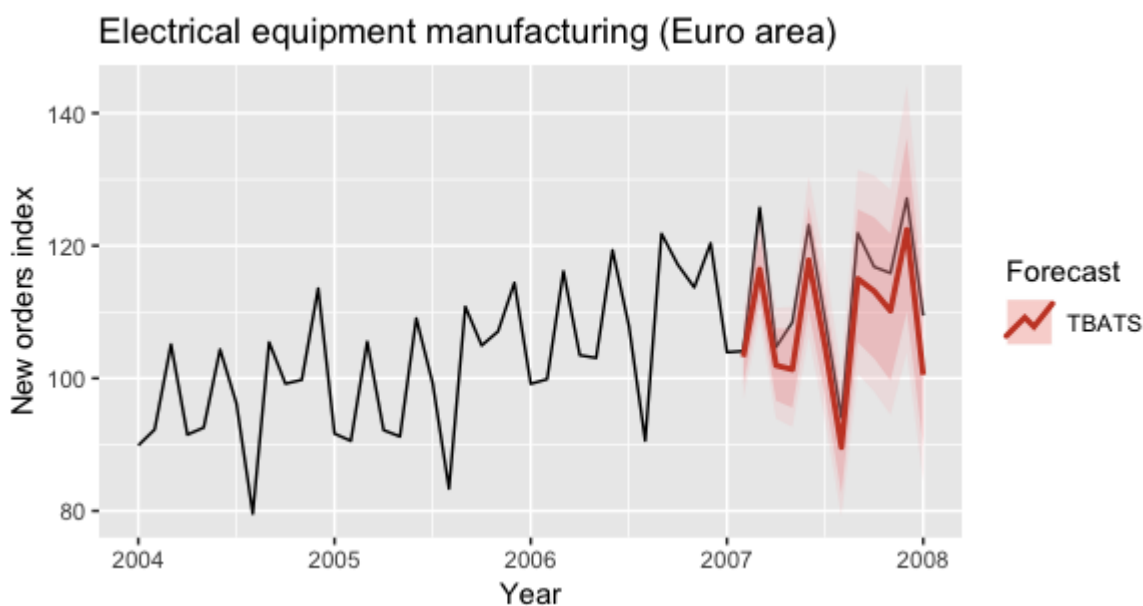


Рис. 2.9. Прогнози, отримані за допомогою моделі TBATS

2.3.8 Модель Пророк

Prophet – це ще одна модель прогнозування, яка дозволяє мати справу з кількома сезонними обставинами. Це програмне забезпечення з відкритим кодом, випущене командою Facebook Core Data Science. Модель пророка передбачає, що часовий ряд можна розкласти наступним чином:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon(t) \quad (2.9)$$

Три змінні $g(t)$, $s(t)$ і $h(t)$ відповідають відповідно тенденції, сезонності та свята. Остання змінна є терміном помилки. Є два варіанти часових рядів тенденцій: модель насиченого зростання та шматково-лінійна модель. Багатоперіодна модель сезонності спирається на ряди Фур'є. Ефект відомих і нестандартних свят можна легко включити в модель. Модель пророка вставлена в байєсівську структуру і дозволяє зробити повний апостеріорний висновок, щоб включити невизначеність параметра моделі в невизначеність прогнозу. Наступний графік (Рис.2.10) показує прогнози, отримані на 2007 рік за допомогою моделі Пророка, щоб відповідати часовим рядам.

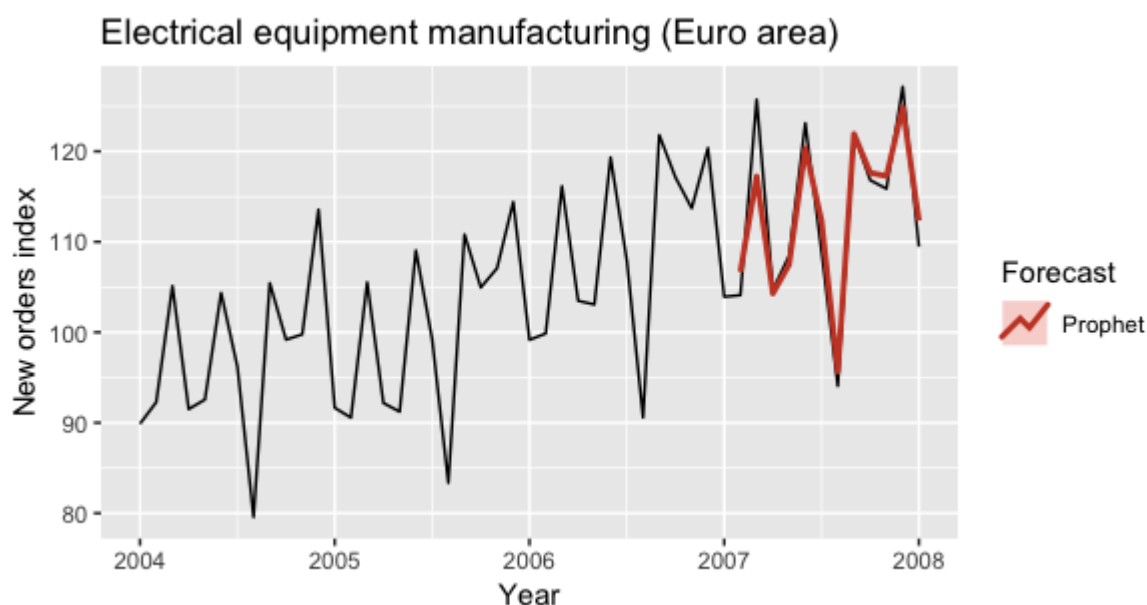


Рис. 2.10. Прогнози, отримані за допомогою моделі Пророка

2.3.9 NNETAR

Модель NNETAR — це повноцінна нейронна мережа. Акронім розшифровується як Neural NETwork AutoRegression. Модель NNETAR приймає на вхід останні елементи послідовності до моменту t і виводить прогнозоване значення в момент $t+1$. Для виконання багатокрокових прогнозів мережа застосовується ітеративно. За наявності сезонності вхідні дані можуть включати

також тимчасові ряди з сезонним відставанням. Наступні графіки (Рис.2.11) показують прогнози, отримані на 2007 рік, отримані з використанням моделі NNETAR із сезонним відставанням та моделі NNETAR для сезонно скоригованого часового ряду.

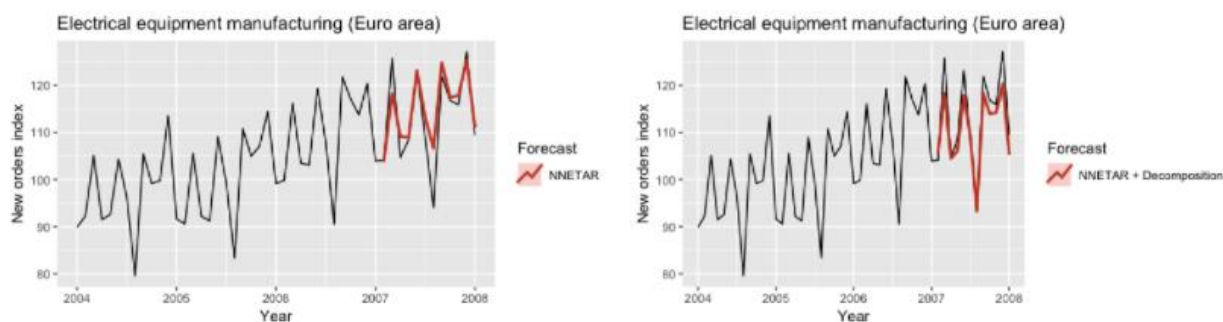


Рис. 2.11. Прогнози, отримані з використанням моделі NNETAR

2.3.10 LSTM

Моделі LSTM можна використовувати для прогнозування часових рядів (а також інших рекурентних нейронних мереж). LSTM — це абревіатура, що означає «довготривалі спогади». Стан мережі LSTM представляється через вектор простору станів. Ця методика дозволяє відстежувати залежності нових спостережень від минулих (навіть дуже далеких).

Загалом, LSTM є складною моделлю, і вона рідко використовується для прогнозування окремого часового ряду, оскільки вона вимагає великої кількості даних для оцінки. Однак вона зазвичай використовується, коли потрібні передбачення для великої кількості часових рядів. Наступний графік (Рис.2.12) показує прогнози для першого року в тестовому наборі, отримані шляхом підгонки моделі LSTM на сезонно скоригований часовий ряд.

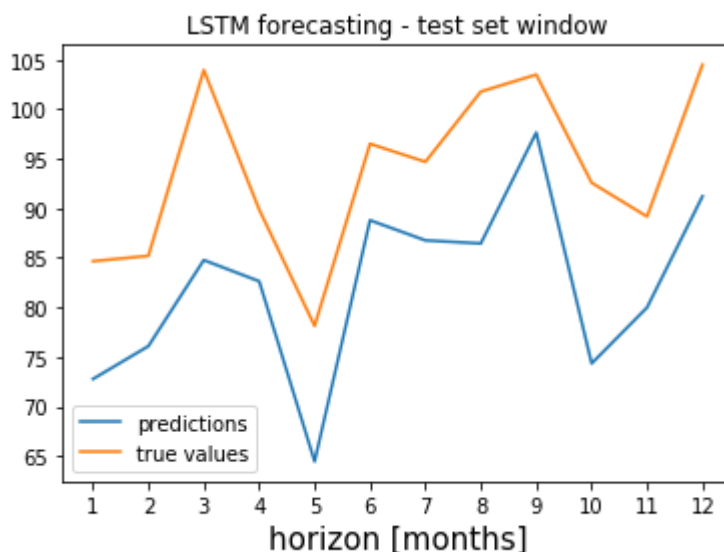


Рис. 2.12. Прогнози для першого року в тестовому наборі, отримані шляхом підгонки моделі LSTM на сезонно скоригований часовий ряд

2.3.11 Оцінка методів

Загальний MAE, обчислений шляхом усереднення за різні часові горизонти виявився наступним для кожної моделі:

- 1) Naïve=11.83;
- 2) Snaïve=8.32;
- 3) Exp Smoothing=5.71;
- 4) SARIMA=5.86;
- 5) ARIMA=5.99;
- 6) GARCH=5.76;
- 7) TBATS=5.77;
- 8) PROPHET=10.41;
- 9) NNETAR=7.82.

Модель Експоненційного згладжування була найкращою моделлю, оскільки вона показала найнижчий перехресний показник MAE.

2.4 Прогнозування за експоненційним згладжуванням

Настав час застосувати вибрану модель для вирішення даної проблеми. Для реалізації даної моделі буде використана мова програмування Python. Було створено скрипт (Рис. 2.13), який відкриває файл з даними (Рис.2.14), у якому містяться показники TDS за три місяці, на основі яких буде прогнозуватись дата, коли TDS перевищить норму.

```

main.py 3
1  from datetime import datetime, timedelta, date
2  import pandas as pd
3  import numpy as np
4  import matplotlib.pyplot as plt
5  get_ipython().run_line_magic('matplotlib', 'inline')
6  import chart_studio.plotly as py
7  import plotly.offline as pyoff
8  import plotly.graph_objs as go
9  df = pd.read_csv("D:\Test\Python\tds.csv")
10 df['Date'] = pd.to_datetime(df['Date'])
11 df.index.freq = 'MS'
12 df
13 train_data = df.iloc[:120]
14 test_data = df.iloc[120:]
15 train_data.info()
16 import plotly.graph_objs as go
17 import chart_studio.plotly as py
18 plot_data = [
19     go.Scatter(
20         x=train_data['Date'],
21         y=train_data['TDS'],
22     )
23 ]
24 plot_layout = go.Layout(
25     title='Train Analysis Chart'
26 )
27 fig = go.Figure(data=plot_data, layout=plot_layout)
28 pyoff.iplot(fig)
29 import plotly.graph_objs as go
30 import chart_studio.plotly as py
31 plot_data = [
32     go.Scatter(
33         x=test_data['Date'],
34         y=test_data['TDS'],
35     )
36 ]
37 plot_layout = go.Layout(
38     title='Test Analysis Chart'
39 )
40 fig = go.Figure(data=plot_data, layout=plot_layout)
41 pyoff.iplot(fig)
42 train_data['TDS'].plot(legend=True, label='TRAIN')
43 test_data['TDS'].plot(legend=True, label='TEST', figsize=(12,8));
44 from statsmodels.tsa.seasonal import seasonal_decompose
45 seasonal_decompose(train_data['TDS'], period=12).plot();

```

Рис. 2.13. Скрипт на мові Python, який реалізує модель експоненційного згладжування

Кафедра інтелектуальних інформаційних систем
Моніторинг та прогнозування параметрів системи очищення води

1	Даты	TDS
2	15.03.2021	11
3	16.03.2021	12
4	17.03.2021	11
5	18.03.2021	12
6	19.03.2021	12
7	20.03.2021	12
8	21.03.2021	11
9	22.03.2021	12
10	23.03.2021	12
11	24.03.2021	12
12	25.03.2021	12
13	26.03.2021	12
14	27.03.2021	12
15	28.03.2021	12
16	29.03.2021	12
17	30.03.2021	12
18	31.03.2021	12
19	01.04.2021	12
20	02.04.2021	13
21	03.04.2021	13
22	04.04.2021	13
23	05.04.2021	12

Рис. 2.14. Дані з фільтру для роботи моделі

У результаті роботи скрипта, було отримано дані (рис.2.15), за якими можна зробити висновок, що картриджі для фільтру можна використовувати, поки кількість літрів не перевищує 7700 літрів.

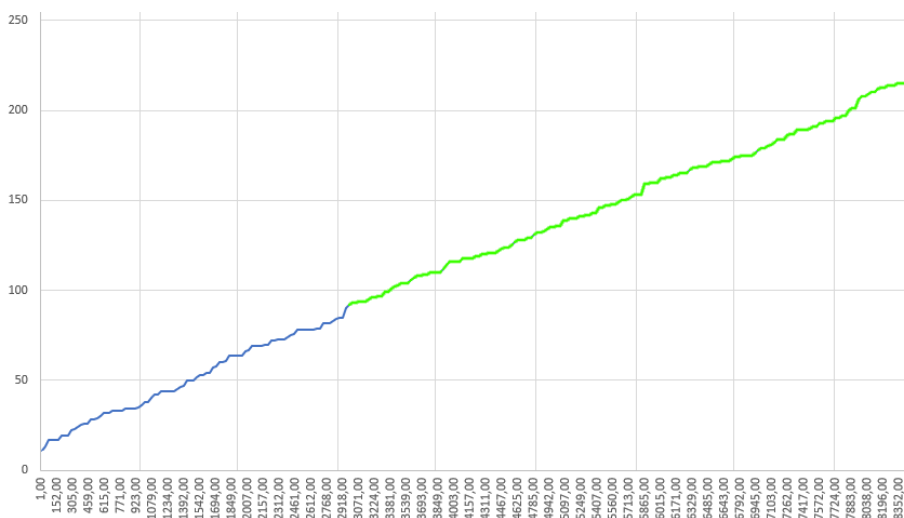


Рис. 2.15. Результат роботи скрипта

Висновки до розділу 2

У даному розділі було описано, що таке прогнозування – це процес передбачення майбутнього стану об'єкта або явища на основі аналізу його минулого і сьогодення, систематичної оцінки інформації про якісні та кількісні характеристики майбутнього розвитку вибраного об'єкта або явища. Результатом прогнозування є прогноз - розуміння майбутнього і ймовірного розвитку поточних тенденцій для конкретного феноменологічного об'єкта в майбутньому.

Також було розглянуто моделі і види прогнозування. Найкращим варіантом для прогнозування кількості літрів як ресурсу картриджа було обрано прогнозування за експоненційним згладжуванням, що в результаті дало показник ресурсу картриджів у ~ 7700 літрів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка програмного забезпечення для системи моніторингу води

Розробка ПЗ для Arduino буде проходити на ПК під керуванням ОС Windows 10 Pro 20H2 x64. Для створення та компіляції коду буде використовуватись Arduino IDE 1.8.15 (рис. 3.1). Платою для компіляції буде Espressif ESP32 devkit ver (рис. 3.2).

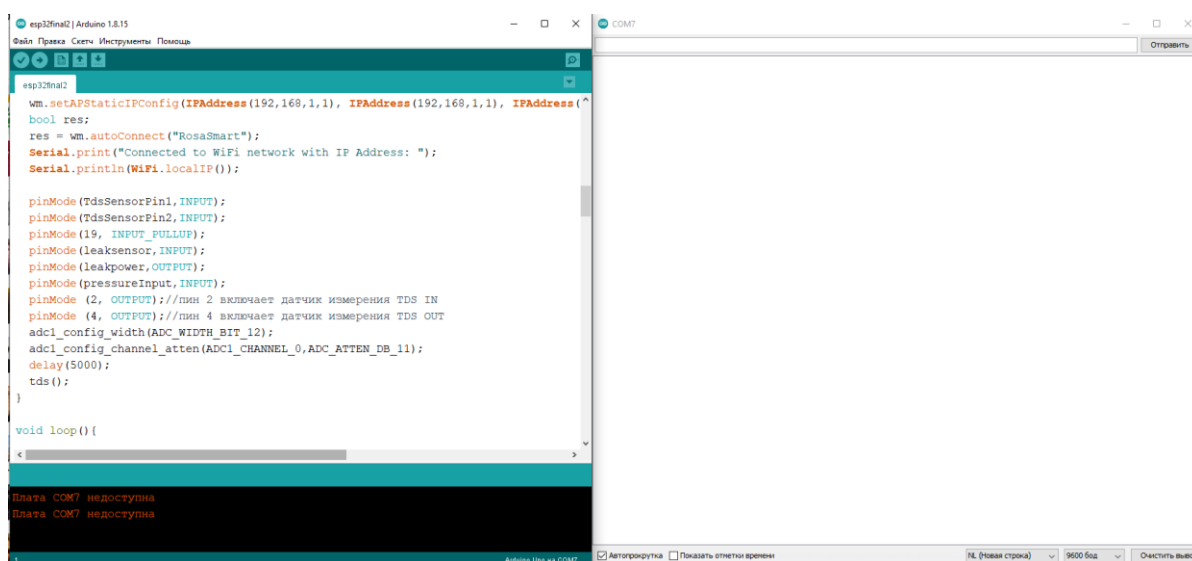


Рис. 3.1. Робоче середовище Arduino IDE 1.8.15

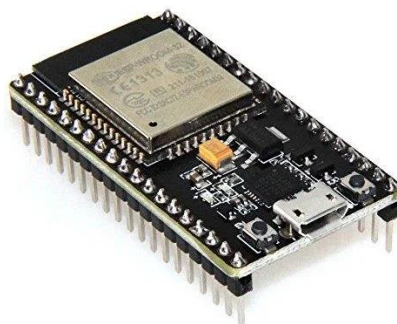


Рис. 3.2. ESP32

Для початку розробки треба підготувати сервер, який буде приймати дані від ардуіно і відправляти їх у БД, для подальшої обробки, зберігання та виводу.

3.2 Налаштування серверу

Для початку треба придбати хостинг. Також, можна використовувати безкоштовні варіанти. Для даного проекту було обрано один з кращих вітчизняних хостингів Hostiq.ua. Після того, як доступ до хостингу отримано, відкриваємо домашній каталог. Спочатку, потрібно буде створити два файли – один для обробки даних, що будуть надходити від ардуіно, та запису їх у БД, а інший – для їх відображення. Перший файл буде мати назву post-esp-data.php (рис. 3.3).

```

$servername = "localhost";

// REPLACE with your Database name
$dbname = "██████████";
// REPLACE with Database user
$username = "██████████";
// REPLACE with Database user password
$password = "██████████";

$api_key= $flowspeed = $liters = $temperature = $pressure = $tds_in = $tds_out = $leak = $ostatok_l = $ostatok_d = $reset = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    $flowspeed = test_input($_POST["flowspeed"]);
    $liters = test_input($_POST["liters"]);
    $temperature = test_input($_POST["temperature"]);
    $pressure = test_input($_POST["pressure"]);
    $tds_in = test_input($_POST["tds_in"]);
    $tds_out = test_input($_POST["tds_out"]);
    $leak = test_input($_POST["leak"]);
    $reset = test_input($_POST["reset"]);

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
}

```

Рис. 3.3. Синтаксис файлу для отримання інформації

У даному файлі, після підключення до БД, оголошуються усі змінні, та записуються у БД. Для передачі даних з Arduino до серверу буде використовуватись метод відправлення даних POST. Також було створено обробник скидання системи – це потрібно для того, щоб при натисканні кнопки на Ардуіно для скидання (зміни картриджу), застарілі дані скидались і на сервері. Для цього було створено умову if, яка при отриманні змінної reset = 1 скидає усі дані, и встановлює початкові показники ресурсу (рис. 3.4).

```

if ($reset == 0) {
    $sql = "INSERT INTO SensorData (api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, ost
    VALUES (" . $api_key . ", " . $flowspeed . ", " . $liters . ", " . $temperature . ", " . $pressure . ",
            " . $tds_in . ", " . $tds_out . ", " . $leak . ", (SELECT MIN(S.ostatok_1) - " . $liters . " FROM
}
else {
    $sql = "DELETE FROM SensorData WHERE api = ".$api_key."; INSERT INTO SensorData (api, flowspeed, liters,
    VALUES (" . $api_key . ", " . $flowspeed . ", 0, " . $temperature . ", " . $pressure . ", " . $tds_in . "
}

```

Рис. 3.4. Обробка скидання

Далі, створимо простий файл для відображення результатів моніторингу. Його задачею буде відвантаження даних з БД, и вивід їх на сторінку браузера користувача (рис. 3.5).

```

echo '<table cellspacing="5" cellpadding="5">
<tr>
<td>api</td>
<td>flowspeed(л/мин)</td>
<td>temperature</td>
<td>pressure</td>
<td>tds_in</td>
<td>tds_out</td>
<td>leak</td>
<td>Timestamp</td>
<td>ostatok_l</td>
<td>ostatok_d</td>
</tr>';

if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_api = $row["api"];
        $row_flowspeed = $row["flowspeed"];
        $row_liters = $row["liters"];
        $row_temperature = $row["temperature"];
        $row_pressure = $row["pressure"];
        $row_tds_in = $row["tds_in"];
        $row_tds_out = $row["tds_out"];
        $row_leak = $row["leak"];
        $row_reading_time = $row["reading_time"];
        $row_ostatok_l = $row["ostatok_l"];
        $row_ostatok_d = $row["ostatok_d"];
        $row_ostatok_lit = number_format($row_ostatok_l, 0, '.', '');
        $row_temperature1 = number_format($row_temperature, 1, '.', '');
        $row_pressure1 = number_format($row_pressure, 1, '.', '');

        echo '<tr>
<td>' . $row_api . '</td>
<td>' . $row_flowspeed . '</td>
<td>' . $row_temperature1 . '</td>
<td>' . $row_pressure1 . '</td>
<td>' . $row_tds_in . '</td>
<td>' . $row_tds_out . '</td>
<td>' . $row_leak . '</td>
<td>' . $row_reading_time . '</td>
<td>' . $row_ostatok_lit . '</td>
<td>' . $row_ostatok_d . '</td>
</tr>';
    }
    $result->free();
}

```

Рис. 3.5. Файл для виводу даних

Тепер, коли два файли створені, можна створити декілька додаткових. Так як БД має межі за розміром, потрібно якось очищати її. До того ж, якщо у БД буде багато значень, сервер буде працювати повільніше. Для цього створимо файл cleaner.php, який за допомогою crontab буде один раз у три дні запускатись і очищати усі значення, окрім останнього для кожного користувача. Для цього напишемо наступний скрипт для MySQL:

```
$sql = "TRUNCATE TABLE smart2;
INSERT INTO smart2 (api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, reading_time, ostatok_l,
ostatok_d)
SELECT api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, reading_time, MIN(ostatok_l) AS
'ostatok_l', ostatok_d FROM SensorData GROUP BY api;
TRUNCATE TABLE SensorData;
INSERT INTO SensorData (api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, reading_time, ostatok_l,
ostatok_d)
SELECT api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, reading_time, ostatok_l, ostatok_d FROM
smart2;
TRUNCATE TABLE smart2;";
```

Даний скрипт при старті знаходить останні записи для кожного api (назва id користувача), копіює їх у тимчасову таблицю, після цього повністю очищує основну таблицю, і копіює дані до неї.

Також додатково, для рахування днів, було створено скрипт, який кожного дня у 00:00 віднімає 1 день у полі ostatod_d, тим самим було реалізовано простіший лічильник днів.

3.3 Створення ПЗ для Arduino

Перше, з чим будуть стикатись користувачі при вмиканні системи – це її під'єднання до мережі Wi-Fi. Так як вони не матимуть доступу про програмного застосунку та не зможуть перепрограмувати мікроконтроллер, є два варіанти – використати WPS або просто під'єднатись до домашньої мережі за логіном і паролем. Від WPS було вирішено відмовитись через низьку надійність, більшу

складність налаштування та загальну ненадійність технології. То як же користувачі зможуть під'єднатись до своєї мережі? Однією із переваг моделей ESP було те, що її модулі Wi-Fi можуть працювати як у режимі клієнту, так і у режимі AP (Access Point, або точка доступу). За допомогою бібліотеки `WifiManager.h` можна реалізувати наступний сценарій користування – користувач вмикає систему, під'єднується до точки доступу Arduino, на локальній сторінці вводить логін і пароль своєї мережі, і після цього система перезавантажується і під'єднується до мережі користувача. Для цього підключаємо бібліотеку за допомогою команди `#include <WiFiManager.h>`. Дана бібліотека вимагає для роботи бібліотеки “`Wifi.h`” і “`WiFiClient.h`”, тому також підключаємо і їх. Перш ніж перейти до розробки, варто сказати що у мові Arduino є дві основні функції – `setup()` і `loop()`. Функція `setup` потрібна для налаштування ардуіно на роботу – вмикання системних функцій, старту роботи обладнання, тощо. А функція `loop()` потрібна для виконання основного коду. Дана функція працює циклічно і не перестає працювати до моменту знеструмлення системи. Отже, у функцію `setup()` вписуємо наступні рядки (Рис. 3.6).

```
void setup() {

  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFiManager wm;
  wm.setAPStaticIPConfig(IPAddress(192,168,1,1), IPAddress(192,168,1,1),
  bool res;
  res = wm.autoConnect("RosaSmart");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
}
```

Рис. 3.6. Створення об'єкту `wm`

У даній частині коду було запущено серійний вивід даних з частотою 115200 бод. Також було задано статичний IP для ардуіно у режимі точки доступу. Тепер при вмиканні пристрою, користувачі будуть мати змогу під'єднатись до

бездротової мережі із будь-якою заданою назвою. У даному випадку це буде RosaSmart. Після цього потрібно буде перейти по заданій адресі у програмі – це 192.168.1.1. Там на користувача буде чекати сторінка з вибором Wi-Fi мереж (рис. 3.7.).

TeRmiNaTor [signal icon]
OSBORN [signal icon]
Alex-KS [signal icon]
WP_Alstim [signal icon]
SPURS [signal icon]

SSID
TeRmiNaTor
Password
[]

Save
Refresh

No AP set

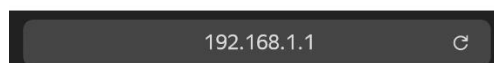


Рис. 3.7. Сторінка вибору бездротових мереж

Далі, коли мережа вибрана, і пароль введений вірно, система перезавантажується, і під'єднується до мережі. Так як логін і пароль останньої точки доступу записується у енергонезалежну пам'ять, при наступних перезавантаженнях система буде пам'ятати вашу ТД і буде під'єднуватись до неї автоматично.

Також у функції setup() потрібно оголосити режим роботи портів з сенсорами (ввод чи вивід) (рис. 3.8).

```
pinMode(19, INPUT_PULLUP);
pinMode(leaksensor, INPUT);
pinMode(pressureInput, INPUT);
pinMode(2, OUTPUT); //пин 2 включає датчик вимірювання TDS IN
pinMode(4, OUTPUT); //пин 4 включає датчик вимірювання TDS OUT
adc1_config_width(ADC_WIDTH_BIT_12);
adc1_config_channel_atten(ADC1_CHANNEL_0, ADC_ATTEN_DB_11);
delay(5000);
```

Рис. 3.8. Оголошення режимів роботи портів

Після цього можна переходити до функції loop(). У ній першим чином рахується швидкість потоку води та літраж (рис.3.9).

```
if (pulseIn(pinSensorf, HIGH, 200000) > 0) {
  signalcount++;
}
liters = signalcount / oneliter;
varPulse = pulseIn(pinSensorf, HIGH, 200000);
if (varPulse) {
  varResult = float(1000000.0 / (15.0 * varPulse)) * 0.250;
}
else {
  varResult = 0.0;
}
```

Рис. 3.9. Розрахунок літражу і швидкості води

У даній області коду відбувається підрахунок кількості імпульсів з датчику потоку води (умовою є довжина імпульсу не більше 0.2 секунди – це зроблено для запобігання ситуацій, коли вода перестає тикти, а датчик зупинився поблизу сенсора, і система вважає що датчик і далі крутиться). Далі за даною формулою прораховується швидкість потоку у літрах за хвилину. Також у даному проміжку обраховується кількість літрів – імпульси діляться на фіксовану змінну, що відповідає за еквівалент літру, але обрахований у імпульсах.

Далі йде функція, яка при умові натискання кнопки скидання, запускає функцію, яка за це відповідає (`resetpacket()`). Далі створено функцію, яка кожні 6 секунд збирає показники усіх сенсорів і датчиків та запускає функцію, яка надсилає отриманні дані до серверу (рис. 3.10).

```

if (millis() - currentTime > 6000.0)
{
    currentTime = millis();
    temp();
    leak();
    tds();
    pressure();
    sendinfo();
}

```

Рис. 3.10. Збір показників та їх надсилання

Далі за формулою $pressure = ((pressureValue - pressureZero) * pressuretransducermaxPSI) / (pressureMax - pressureZero)$; вираховується тиск, де `pressureValue` – показник сенсору тиску, `pressureZero` – показання сенсору при відсутньому тиску, `pressuretransducermaxPSI` – максимальна паспортна кількість тиску у psi (100 psi) (футнів на квадратний дюйм), `pressureMax` - показання сенсору при максимальному (100 psi) тиску.

Для запобігання протoku води, два не з'єднані контакти будуть відходити від системи і лежати на піддлозі під фільтром. Функція `waterleak = analogRead(leaksensor)`; перевіряє проток води – якщо значення сенсору не є 0 – тоді на сервері увімкнеться повідомлення про протік води. Температура води рахується за формулою $temperature = 9000000 * pow(analogRead(tempsensor), -1.672)$;

Наступною функцією є `resetpacket`, яка є ідентичною по наповненню до `sendinfo`, але з однією різницею – змінна `reset` статично встановлена у позицію 1, або `true`.

Останньою функцією є `tds()`, яка виміряє чистоту води на предмет домішок і солей (рис. 3.11).

```
void tds() {
    digitalWrite (2, HIGH);
    delay (500);
    int val1 = adc1_get_raw(ADC1_CHANNEL_0);
    tdsValue1 = (1.47*val1-113.0)-2.0;
    digitalWrite (2, LOW );
    delay (3000);
    digitalWrite (4, HIGH);
    delay (500);
    int val2 = adc1_get_raw(ADC1_CHANNEL_0);
    tdsValue2 = (0.202*val2-211.0)+11;|
    digitalWrite (4, LOW );
    delay(3000);
}
```

Рис. 3.11. Вимірювання вхідного і вихідного рівню tds води

За допомогою функції `adc1_get_raw(ADC1_CHANNEL_0)`; отримуємо більш точні дані із сенсору `tds`. Принцип дії функції наступний – при активації, ардуіно спочатку виміряє вхідний `tds`, а після цього за допомогою подачі напруги до піну 4, перемикає сенсор на вимірювання вихідного `tds` води. Отримані дані обчислюються за формулою, указаною на рис. 3.11.

Далі йде функція `sendinfo()` яка збирає усі отримані дані (у т.ч. номер системи), і за допомогою функції `int httpResponseCode = http.POST(httpRequestData)`; надсилає пакет даних до серверу методом POST і отримує код відповіді (якщо все добре, це буде 200 код).

Висновки до розділу 3

У даному розділі було описано і створено програмне забезпечення для системи, а саме для Arduino та для серверу. Код для мікроконтроллеру запрограмований на зчитування показників усіх сенсорів, модулів та кнопок і подальшу відправку отриманої інформації на сервер. Сервер було сконфігуровано для прийняття, обробки і запису у БД даних і Arduino. Також сервер має змогу виводити отримані дані у структурованому виді у вигляді таблиці.

ВИСНОВКИ

За час виконання даної роботи було вирішено перелік задач:

7) проаналізовано ринок фільтрів для води, знайдено найкращий варіант для подальшого створення системи навколо нього;

8) розглянуто усі доступні мікроконтролери, сформувавши список вимог до них, і на основі порівняння обрано найкращий варіант;

9) ознайомлено з найпопулярнішими мовами програмування і середовищами для програмування Arduino, на основі переваг і недоліків обрано найкращу;

10) обрано типи, види і протоколи зв'язку, що відповідають заданим умовам системи;

11) обрано дані для моніторингу, вид і модель прогнозування;

12) виконано прогнозування ресурсу картриджів для фільтру та розроблено ПЗ для серверу і Arduino.

У результаті виконання усіх даних пунктів було створено сучасну, недорогу та конкурентоспроможну систему, яка має можливості для оновлення, наприклад додавання декількох датчиків, чи зміну джерела живлення або способу зв'язку.

В результаті написання спеціальної частини з охорони праці було досягнуто поставленої мети, а саме створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях.

80%D0%B8_%D0%B4%D0%BB%D1%8F_%D0%B2%D0%BE%D0%B4%D0%B8
(дата звернення: 12.01.2022).

9. Метод експоненційного згладжування : Вебсайт. URL:
https://studme.com.ua/1115121211736/menedzhment/metod_eksponentsialnogo_sglazhivaniya.htm (Дата звернення 12.01.2022).

10. ESP32 with BME280 Sensor using Arduino IDE : Вебсайт. URL:
<https://randomnerdtutorials.com/esp32-bme280-arduino-ide-pressure-temperature-humidity/> (Дата звернення 12.01.2022).

11. Aquarium Water Quality Monitor with TDS Sensor & ESP32 IDE :
Вебсайт. URL: <https://how2electronics.com/aquarium-water-quality-monitor-with-tds-sensor-esp32/> (Дата звернення 12.01.2022).

12. Interface Water Flow Sensor Using ESP32 Board : Вебсайт. URL:
<https://tutorial.cytron.io/2019/10/02/interface-water-flow-sensor-using-esp32-board/>
(Дата звернення 12.01.2022).

13. PHP \$_POST - Manual : Вебсайт. URL:
<https://www.php.net/manual/ru/reserved.variables.post.php> (Дата звернення
12.01.2022).

14. Обработка POST-запросы в PHP: Вебсайт. URL:
<https://php.zone/kurs-php-dlya-nachinayushih/post-zaprosy-v-php> (Дата звернення
12.01.2022).

15. Блум Дж. Вивчаємо Arduino [пер. з англ. Петін В.] - СПб.: БХВ-
Петербург - 2015. 336 с.

16. Голіцина, О.Л. Інформаційні технології: Підручник / О.Л. Голіцина,

17. Н.В. Максимов, Т.Л. Партика, І.І. Попов. - М.: Форум, ИНФРА-М,
2013. - 608с.

18. Амперка : Вебсайт. URL: <http://wiki.amperka.ru> (Дата звернення
14.01.2022).

19. Закон України «Про охорону навколишнього природного середовища» –К.: Україна. – 1991. – 59 с. (з усіма редакціями до 2017 року);
20. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин/Зареєстровано в Міністерстві юстиції України 19 квітня 2010 р. за №293/17588;
21. Правила улаштування електроустановок. ПУЕ.– Харків.: Форт – 2011 – 728 с.;
22. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу. Гігієнічні нормативи ГН 3.3.5-8-6.6.1 2002 р. Видання офіційне Київ, 2001 рік – 46 с.
23. ДБН В.2.5-67:2013. Опалення, вентиляція та кондиціонування. – К.: Мінрегіон України, 2013. – 147 с.;
24. ДБН.В.2.5 – 28-2006. Природне і штучне освітлення. – К.: Мінбуд України, - 2008 – 74 с.;
25. НАПБ Б.03.002 – 2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. Наказ МНС від 03.12.2007 №883;
26. ДСанПін 3.3.2.007– 98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. – К.: ГСЕУ України, 1998 – 21 с.;
27. ДБН В.1.1 – 7- 2002. Захист від пожежі. Пожежна безпека об'єктів будівництва. – К.: 2002. – 41 с.;
28. ДСТУ ISO14001 – 97 – 14012-97. Система управління оточующою середою – К.: Держстандарт України – 225 с.;
29. Arduino programming notebook : веб-сайт. URL: <http://engineering.nyu.edu/gk12/ampscbr/pdf/ArduinoBooks/Arduino%20Programming%20Notebook.pdf> (Дата звернення 18.01.2022).

30. ESP32 with BME280 Sensor using Arduino IDE : Вебсайт. URL: <https://randomnerdtutorials.com/esp32-bme280-arduino-ide-pressure-temperature-humidity/> (Дата звернення 12.01.2022).
31. Aquarium Water Quality Monitor with TDS Sensor & ESP32 IDE : Вебсайт. URL: <https://how2electronics.com/aquarium-water-quality-monitor-with-tds-sensor-esp32/> (Дата звернення 12.01.2022).
32. Interface Water Flow Sensor Using ESP32 Board : Вебсайт. URL: <https://tutorial.cytron.io/2019/10/02/interface-water-flow-sensor-using-esp32-board/> (Дата звернення 12.01.2022).

ДОДАТОК А

ЛІСТИНГ КОДУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

```
#include <Wire.h>
#include <WebServer.h>
#include <WiFiClient.h>
#include <WiFi.h>
#include <WiFiManager.h>
#include <HTTPClient.h>
#include <driver/adc.h>

#define leaksensor 34 //АНАЛОГ ПРОТЕЧКИ
#define pinSensorf 13 //DIGITAL ФЛОУ МЕТЕР
#define tempsensor 35 //АНАЛОГ ТЕМПЕРАТУРЫ

String apiKeyValue = "2"; //Номер системи

unsigned long currentTime = 0;
unsigned long currentTime2 = 0;
unsigned long currentTime3 = 0;

const char* serverName = "http://smart.rosa1.com/post-esp-data.php";

String pressurestring;

int lastState = LOW;
int currentState;

unsigned long pressedTime = 0;
unsigned long releasedTime = 0;
```

```
const int LONG_PRESS_TIME = 3000;

int waterleak = 0;
uint8_t tdsValue1 = 0, tdsValue2 = 0;
float temperature = 25.0;
float davlenieR;
float davltest;
float dav1;
float dav2;
float dav3;
float dav4;
float dav5;

int chetchik = 0;

const int pressureInput = 39; //АНАЛОГ ДАВЛЕНИЯ
float davlenie;
uint32_t varPulse; // Объявляем переменную для хранения
длительности импульсов с датчика
float varResult;
float flowRate = 0.0;
unsigned int flowMilliLitres = 0;
unsigned long totalMilliLitres = 0;
unsigned long oldTime = 0;
unsigned long signalcount = 0; //вернуть лонг
float liters;
float oneliter = 850.0; //НАСТРОЙКА СЕНСОРА ВОДЫ
int eeAddress = 0;
```

```
int button_state;

void setup() {

  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFiManager wm;
  wm.setAPStaticIPConfig(IPAddress(192,168,1,1),      IPAddress(192,168,1,1),
  IPAddress(255,255,255,0));
  bool res;
  res = wm.autoConnect("RosaSmart");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  pinMode(19, INPUT_PULLUP);
  pinMode(leaksensor,INPUT);
  pinMode(pressureInput,INPUT);
  pinMode (2, OUTPUT);//пин 2 включає датчик вимірювання TDS IN
  pinMode (4, OUTPUT);//пин 4 включає датчик вимірювання TDS OUT
  adc1_config_width(ADC_WIDTH_BIT_12);
  adc1_config_channel_atten(ADC1_CHANNEL_0,ADC_ATTEN_DB_11);
  delay(5000);
  tds();
}

void loop(){
```

```
    if (pulseIn(pinSensorf, HIGH, 200000) > 0) {
signalcount++;
}
Serial.print("кол-во сигналів на датчику холла ");
Serial.println(signalcount);
liters = signalcount / oneliter;
Serial.print("літров: ");
Serial.print(liters);
Serial.println("");
varPulse = pulseIn(pinSensorf, HIGH, 200000);
if (varPulse) {
    varResult = float(1000000.0/(15.0*varPulse))*0.250;
}
else {
    varResult = 0.0;
}
Serial.println((String) "СКОРОСТЬ = " + varResult + " L/MIN");
Serial.println(digitalRead(19));
currentState = digitalRead(19);
if (currentState != HIGH)
{
    Serial.println("СБРОС");
    signalcount = 0;
    liters = 0.0;
    resetpacket();
}
```

```
// if ( ( millis() - currentTime2 > 10000.0) && varResult == 0.0 )      //тдс
каждые 5 минут
if ((millis() - currentTime2 > 10000.0))      //тдс каждые 5 минут
{
  currentTime2 = millis();
  tds();
}

if (millis() - currentTime > 6000.0)      //время отправки пакетов 6 секунд
{
  currentTime = millis();
  temp();
  leak();
  sendinfo();
}

if (millis() - currentTime3 > 2000.0)      //время отправки пакетов 6 секунд
{
  currentTime3 = millis();
  pressure();
}

}

void sendinfo(){

if(WiFi.status()== WL_CONNECTED){
  WiFiClient client;
```

```
HTTPClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
String httpRequestData = "api_key=" + apiKeyValue + "&flowspeed=" +
varResult
    + "&liters=" + liters + "&temperature=" + temperature
    + "&pressure=" + davltest + "&tds_in=" + tdsValue2 + "&tds_out="
+ tdsValue1 + "&leak=" + waterleak + "&reset=0";
Serial.print("httpRequestData: ");
Serial.println(httpRequestData);
int httpResponseCode = http.POST(httpRequestData);
if (httpResponseCode==200) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    signalcount = 0;
    liters = 0.0;
}
else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
http.end();
}
else {
    Serial.println("WiFi Disconnected");
    WiFiManager wm;
    bool res = wm.autoConnect("RosaSmart");
}
```

```
}
```

```
void pressure()
```

```
{
```

```
    davlenie = analogRead(pressureInput);
```

```
    switch (chetchik) {
```

```
        case 0:
```

```
            dav1 = (davlenie - 1535) * 0.0656;
```

```
            chetchik = 1;
```

```
            break;
```

```
        case 1:
```

```
            dav2 = (davlenie - 1535) * 0.0656;
```

```
            chetchik = 2;
```

```
            break;
```

```
        case 2:
```

```
            dav3 = (davlenie - 1535) * 0.0656;
```

```
            chetchik = 3;
```

```
            break;
```

```
        case 3:
```

```
            dav4 = (davlenie - 1535) * 0.0656;
```

```
chetchik = 4;
break;

case 4:
dav5 = (davlenie - 1535) * 0.0656;
chetchik = 0;
break;
}

davltest = ((dav1+dav2+dav3+dav4+dav5)/5.0) - 2.0;

}

void leak ()
{
    waterleak = analogRead(leaksensor);
}

void temp ()
{
    temperature = 9000000*pow(analogRead(tempsensor), -1.672);
}

void resetpacket(){

    if(WiFi.status()== WL_CONNECTED){
        WiFiClient client;
```



```
HTTPClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
String httpRequestData = "api_key=" + apiKeyValue + "&flowspeed=" +
varResult
    + "&liters=0&temperature=" + temperature
    + "&pressure=" + davlenieR + "&tds_in=" + tdsValue1 +
"&tds_out=" + tdsValue2 + "&leak=" + waterleak + "&reset=1";
Serial.print("httpRequestData: ");
Serial.println(httpRequestData);
int httpResponseCode = http.POST(httpRequestData);
if (httpResponseCode==200) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    signalcount = 0;
    liters = 0.0;
}
else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
http.end();
}
else {
    Serial.println("WiFi Disconnected");
    WiFiManager wm;
    bool res = wm.autoConnect("RosaSmart");
}
```

```
}
```

```
void tds() {  
    digitalWrite (2, HIGH);  
    delay (500);  
    Serial.print("TDSIN=");  
    int val1 = adc1_get_raw(ADC1_CHANNEL_0);  
    tdsValue1 = (1.47*val1-113.0)-2.0;  
    Serial.println(tdsValue1); //получаем значение с пина 36  
    Serial.print("Raw data in: ");  
    Serial.println(val1);  
    digitalWrite (2, LOW );  
    delay (3000);  
    digitalWrite (4, HIGH);  
    delay (500);  
    Serial.print("TDSOUT=");  
    int val2 = adc1_get_raw(ADC1_CHANNEL_0);  
    tdsValue2 = (0.202*val2-211.0)+11;  
    Serial.println(tdsValue2); //получаем значение с пина 36  
    Serial.print("Raw data out: ");  
    Serial.println(val2);  
    digitalWrite (4, LOW );  
    delay(3000);  
}
```

ДОДАТОК Б

ЛІСТИНГ КОДУ ОБРОБКИ ДАНИХ

```
$servername = "localhost";  
// REPLACE with your Database name  
$dbname = "*****";  
// REPLACE with Database user  
$username = "*****";  
// REPLACE with Database user password  
$password = "*****";  
$api_key= $flowspeed = $liters = $temperature = $pressure = $tds_in = $tds_out = $leak  
= $ostatok_1 = $ostatok_d = $reset = "";  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $api_key = test_input($_POST["api_key"]);  
    $flowspeed = test_input($_POST["flowspeed"]);  
    $liters = test_input($_POST["liters"]);  
    $temperature = test_input($_POST["temperature"]);  
    $pressure = test_input($_POST["pressure"]);  
    $tds_in = test_input($_POST["tds_in"]);  
        $tds_out = test_input($_POST["tds_out"]);  
        $leak = test_input($_POST["leak"]);  
        $reset = test_input($_POST["reset"]);  
  
    // Create connection  
    $conn = new mysqli($servername, $username, $password, $dbname);  
    // Check connection  
    if ($conn->connect_error) {  
        die("Connection failed: " . $conn->connect_error);  
    }  
}
```

```

}
if ($reset == 0) {
    $sql = "INSERT INTO SensorData (api, flowspeed, liters, temperature, pressure,
tds_in, tds_out, leak, ostatok_l, ostatok_d)
    VALUES (" . $api_key . ", " . $flowspeed . ", " . $liters . ", " . $temperature . ", " .
$pressure . ",
        " . $tds_in . ", " . $tds_out . ", " . $leak . ", (SELECT MIN(S.ostatok_l) - "
. $liters . " FROM SensorData AS S WHERE api = " . $api_key . "), (SELECT
MIN(P.ostatok_d) FROM SensorData AS P WHERE api = " . $api_key . "))";
    }
    else {
        $sql = "DELETE FROM SensorData WHERE api = ".$api_key."; INSERT INTO
SensorData (api, flowspeed, liters, temperature, pressure, tds_in, tds_out, leak, ostatok_l,
ostatok_d)
        VALUES (" . $api_key . ", " . $flowspeed . ", 0, " . $temperature . ", " . $pressure
. ", " . $tds_in . ", " . $tds_out . ", " . $leak . ", 6000, 180)";
    }

    if ($conn->multi_query($sql) == TRUE) {
        echo "New record created successfully";
    }
    else {
        echo "Error pushData: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();

```

```
}  
else {  
    echo "No data posted with HTTP POST."  
}
```

```
function test_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```