

Система Передавання Даних На Базі Мережевих Протоколів Зв'язку Для Іот

Нікольський Сергій

Кафедра обчислювальної техніки
Національний технічний університет
України «Київський політехнічний
інститут імені Ігоря Сікорського»

Київ, Україна

Orcid id: 0000-0003-4893-3339

Сергій Ванярха

Кафедра обчислювальної техніки
Національний технічний університет
України «Київський політехнічний
інститут імені Ігоря Сікорського»

Київ, Україна

Orcid id: 0000-0003-2824-160X

Ірина Клименко

Кафедра обчислювальної техніки
Національний технічний університет
України «Київський політехнічний
інститут імені Ігоря Сікорського»

Київ, Україна

Orcid id: 0000-0001-5345-8806

Анотація — Стаття присвячена актуальним питанням розвитку архітектури систем розумних речей і засобів їх взаємодії на міжсерверному та апаратному рівнях. У статті запропоновано архітектурне рішення та програмно-апаратна реалізація для забезпечення процесу передачі даних на основі мережевих протоколів зв'язку для мікроконтролерів, що не мають вбудованих засобів підключення до мережі.

Ключові слова — Інтернет речей, IoT, Wi-Fi, вбудовані розумні системи

I. ВСТУП

Продуктивність сучасних обчислювальних пристроїв дорівнює або навіть перевершує електронно-обчислювальні машини останніх поколінь, при цьому маючи вкрай невеликі розміри і низьку вартість. Разом з цим їх широка доступність і низька вартість засобів зв'язку сприяли розвитку мобільних технологій і тепер кожен володіє невеликим власним суперкомп'ютером, що лежить у нього в кишені. Розвиток інтернету та мережевих протоколів зв'язку породило глобальні мережі обчислювальних пристроїв, які зараз відомі, як технологія IoT. IoT - це нова ера розподілених обчислювальних пристроїв, яка покликана забезпечувати комфорт і безпеку її користувачам.

На сьогодні розвиток IoT є актуальним і масовим напрямком науково-прикладних досліджень у всьому світі. Багато компаній-розробників бере участь у цьому процесі [1], їх задача – створення комфортних та зручних систем для користувача, автоматизація непотрібних рутинних справ, спрощення життя людини, прискорення комунікації, прискорення часу обробки даних, надійність та захист інформації.

В статті пропонується архітектурне рішення та програмно-апаратна реалізація для забезпечення процесу передачі даних на основі мережевих протоколів зв'язку для мікроконтролерів, що не мають вбудованих засобів підключення до мережі.

В якості прикладу практичного застосування запропонованих засобів реалізовано засоби візуалізації параметрів системи на картах «Google

Maps». Запропоноване в статті рішення орієнтоване на збирання та оброблення різноманітної інформації для систем SmartCity.

II. ПОСТАНОВКА ЗАВДАННЯ ТА ОПИС АРХІТЕКТУРИ СИСТЕМИ

На сьогодні всі рішення комунікації та передачі даних від локальної точки до серверу є закритими та приватизованими компаніями та корпораціями. Завдання створення відкритої системою є привабливим і актуальним, оскільки не було знайдено жодного публічного рішення, доступного для доопрацювання користувачем, який бажає створити власні IoT рішення.

Для створення системи комунікації між мікроконтролером та мережевими засобами передавання даних загалом постає задача реалізації спеціалізованого інтерфейсу для підключення до глобальної мережі передавання даних. В якості такого інтерфейсу доцільно використовувати стандартні Wi-Fi модулі, які вже мають підтримку стеку мережевих протоколів.

Розроблювані засоби повинні надати можливість мікроконтролеру приєднатися до Wi-Fi мережі за визначеними користувачем параметрами зафіксувати адрес мережі та надіслати її у вигляді тестових даних на хмарний сервер. Система може використовувати хмарні сервіси, наприклад, компанії Google для визначення місцезнаходження абонента або датчика зафіксувати час під'єднання у вигляді геолокації на карті [2].

Популярні плати Arduino вже мають вбудовані інтерфейси для підтримки мережевих засобів, але сімейства мікроконтролерів установлених на цих платах мають доволі обмежені функціональні можливості і не розглядаються на сьогодні, як технологія для реалізації систем IoT [3].

Для реалізації системи IoT ми застосували мікроконтролер від компанії STMicroelectronics. Даний мікроконтролер належить до сімейства STM32 – це сімейство мікроконтролерів архітектури ARM Cortex-M, розроблених спеціально для реалізації сучасних потужних мереж розумних пристроїв. Даний мікроконтролер переважає усі доступні аналоги за обчислювальними можливостями, розмірами,

гнучкістю, тощо, за невеликою вартістю і доступністю. Мікроконтролер поставляється у комплекті зі StarterKit компанії Global Logic Ukraine [4]. Комплекс представляє собою плату STM32F4DISCOVERY і плату розширення, розроблену компанією Global Logic Ukraine. Серед модельного ряду STM32 серія «Discovery» є найбільш розповсюдженою. Модель STM32F407G-DISC є найсучаснішим представником цієї серії мікроконтролерів [5]. На платі STM32F4DISCOVERY відсутні вбудовані мережеві інтерфейси, таким чином завдання поставлене в статті є актуальним.

Для вибору пристрою комунікації між мікроконтролером та віддаленим сервером було проаналізовано серію рішень та пристроїв. Лідером по функціональним можливостям та простоті використання визначено Wi-Fi модуль ESP-01S, на базі мікроконтролеру ESP-8266 компанії Espressif Systems [5]. Даний Wi-Fi модуль на базі мікроконтролера ESP8266 дозволяє приєднатися до будь-якого мікроконтролера, що підтримує технологію USART. Для підключення використовуються два контакти, один з яких відповідає за відправку даних – (TX, Transceive), а другий слухає канал отримання даних (RX, Receive). Для живлення приладу використовується окремий контакт для живлення із напругою 3.3 Вольти.

Для взаємодії із пристроєм використовується система AT-команд, яка раніше була розроблена для протоколу взаємодії із телефонними мікропроцесорами, а також для взаємодії між телефоном та GSM станцією [6].

III. ОРГАНІЗАЦІЯ ХМАРНОГО СЕРВЕРУ

Для реалізації системи передавання даних на базі мережевих протоколів був розроблений TCP сервер, до якого приєднується мікроконтролер для передавання даних. У якості хмарного серверу був розроблений та піднятий сервер на операційній системі Debian, з власним убудованим TCP-сервером. Для встановлення з'єднання із сервером між клієнтом та сервером створено сокет з'єднання. Для забезпечення безпечності передавання даних та зберігання ресурсів, після відправки пакету даних з'єднання деактивується [7].

IV. ДОПОМІЖНІ КОМПОНЕНТИ СИСТЕМИ

Для визначення місцезнаходження абонента використовується спеціальна API-взаємодія із сервісом GEOLOCATIONS компанії Google, що дозволяє визначати місцезнаходження Wi-Fi мережі за визначеними параметрами. У якості бази даних використовується CSV файл. Такий тип бази даних було обрано з причини високої швидкої. Читання та запис у базу даних відбуваються практично миттєво. Експерименти показали позитивні результати для невеликих проектів та систем, якими можуть бути системи з кінцевою кількістю датчиків для реалізації певного функціоналу. Звісно це не ідеальне рішення, якщо враховувати масштабованість систем і

підтримку до 1-го мільйона абонентів. Для такої кількості звернень CSV-файли є неефективними і варто звернути увагу на більш масштабовані бази даних, такі як SQL, POSTGRESS, тощо.

Для виводу даних ми використали API та сервіс від компанії Google під назвою Google Maps. Було згенеровано приватний API ключ, який надалі використовувався у написаному розробниками компанії Google сервісі написаному на мові програмування Python.

V. АЛГОРИТМ РОБОТИ СИСТЕМИ

Нами було розроблено два алгоритми роботи підсистем для мікроконтролера та сервера, оскільки їх функціонування є незалежним.

Алгоритм функціонування мікроконтролера:

1. Запуск мікроконтролера;
2. Аналіз прошитих даних – для забезпечення безпеки;
3. Ініціалізація підсистем;
4. Сканування доступних Wi-Fi мереж;
5. Під'єднання до мережі;
6. Спроба зв'язатися із віддаленим сервером;
7. У разі успіху – надсилання даних;
8. Деактивація з'єднання до виникнення наступної потреби надсилання даних.

Алгоритм функціонування серверу:

1. Запуск серверу;
2. Ініціалізація допоміжних систем;
3. Створення черги;
4. При надходженні сокет з'єднання – постановка в чергу;
5. Аналіз надісланих даних у порядку черги серверними процесорами та розміщення інформації у базі даних;
6. Виклик допоміжної підсистеми геолокації та пересилання даних;
7. Виклик допоміжної підсистеми геокарти та розміщення пунктів геолокаційних позицій;
8. Видалення із черги опрацьованої задачі.

VI. ДЕМОНСТРАЦІЯ РЕЗУЛЬТАТІВ

Програмно-апаратне забезпечення для реалізації системи передавання даних на базі мережевих протоколів, що складається із програми мікроконтролера, Wi-Fi модуля, TCP сервера, програм серверної частини та додатків до серверної частини були розроблені та протестовані.

На рис. 1 показано фрагмент лог-файлу серверу, на якому було здійснено запуск сервера.

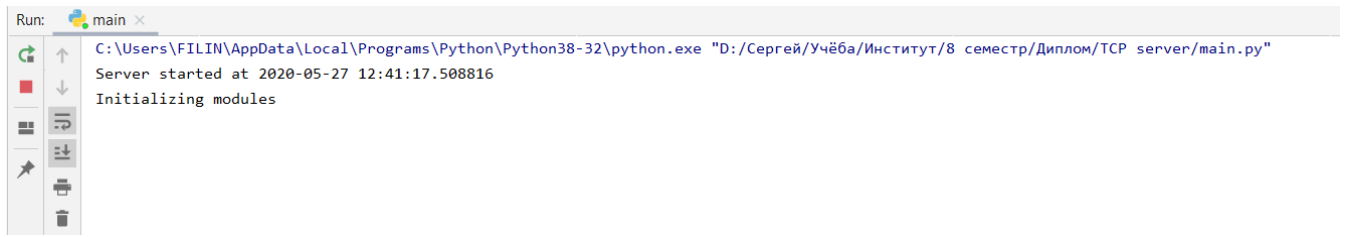


Рис. 1. Лог-файл серверу

Після запуску мікроконтролера відбувається проходження по алгоритму команд, з'єднання та надсилання пакету даних, що фіксується у «логері» мікроконтролера. Після відправки пакету мікроконтролер від'єднується від серверу. Частина «логера», що стосується надсилання даних продемонстрована на рис. 2.

Рис. 2. Лог-файл мікроконтролера

На стороні серверу при надходженні даних відбувається прийняття з'єднання, розміщення клієнтів у черзі, та їх поступова обробка у порядку цієї черги. У процесі обробки зберігаються дані до бази даних та виводяться на мапу, використовуючи додаткові сервіси. Місцезположення визначається за допомогою API GEOLOCATIONS від компанії Google. Вивід даних на карту відбувається за допомогою API GOOGLMAPS від компанії Google. На рис. 3 продемонстрований лог-файл з'єднання, обробки даних та відімкнення клієнту від сервера. В ході тестування тестова база була заповнена тестовими значеннями встановленими емпірично із різних точок Wi-Fi доступу.

```
Run: main x
C:\Users\FILIN\AppData\Local\Programs\Python\Python38-32\python.exe
Server started at 2020-05-27 12:47:39.222387
Initializing modules
Connected by ('100.81.208.29', 49850)
('100.81.208.29', 49850) 2020-05-27 12:57:49.220547
disconnect
```

Рис. 3. Лог-файл з'єднання, обробки даних та відімкнення клієнту від сервера

Приклад бази даних наведено на рис. 4.

Згідно рис 4:

- Перша колонка – номер клієнтського з'єднання, яке визначається із мак адресу мережі;
- Друга колонка – ціна користувача, була введена і добавлена на випадок комерційного використання системи, у випадку відсутності ціни встановлюється нуль;
- Третя колонка – відповідає за час, у який відбувалося з'єднання та обробка результатів;
- Четверта колонка – IP адреса користувача, вона є суттєвою і необхідною для провайдерів мережі, таким чином вони можуть візуалізувати місцерозташування IP адресів своїх клієнтів для кращого усунення технічних проблем;
- П'ята колонка – координата довготи геолокації;
- Шоста колонка – координата ширини геолокації.

1	price	time	client_addr	lon	lat
2	43209,0	2020-05-20 18:49:37.219379	('192.168.1.104', 43209)	30.451302400000003	50.4496128
3	49849,0	2020-05-20 19:12:23.454702	('192.168.1.104', 49849)	30.45336614406	50.44754893794
4	41984,0	2020-05-20 19:18:40.580250	('192.168.1.104', 41984)	30.45336614406	50.44754893794
5	10703,0	2020-05-20 19:19:54.391725	('192.168.1.104', 10703)	30.45336614406	50.44754893794
6	10704,0	2020-05-27 12:57:49.220547	('100.81.208.29', 49850)	33.45208711351	47.9569026752

Рис. 4. Приклад записів в базі даних

Одразу після отримання даних додатком формується оновлення карти, яке відразу стає доступним і розташовується у директорії каталогу під назвою server_map.html, приклад продемонстровано на рис. 5. На мапі в той же момент часу згенеровано та виведено отримані дані. Для тесту мікроконтролер був запущений із Кривого Рогу, що продемонстровано на рис. 6.

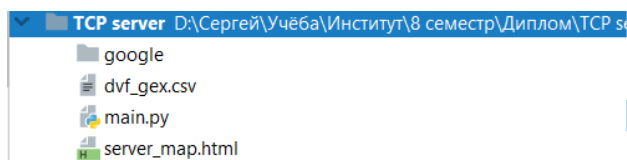


Рис. 5. Файлова структура проєкту серверу

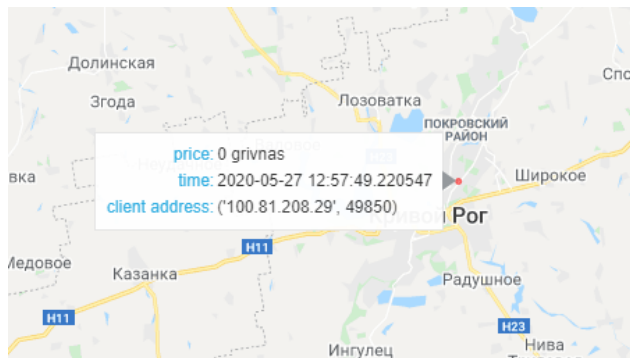


Рис. 6. Демонстрація роботи мікроконтролеру у Кривому Розі

Також, як на рис. 4, виписки бази даних, присутні й інші тестові заміри роботи системи. Можна помітити дві позиції поблизу станції метро Політехнічний інститут на рис 7.

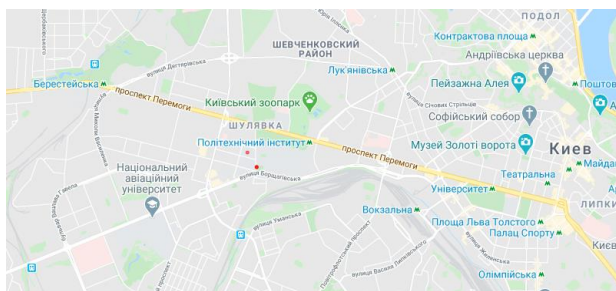


Рис. 7. Демонстрація роботи мікроконтролеру у Києві

VII. ОСОБЛИВОСТІ НАЛАШТУВАННЯ СИСТЕМИ

Створена система доступна для розгортання користувачем без значних труднощів у реалізації та експлуатації. Основними перевагами системи є сучасність, оскільки подібних відкритих аналогів не існує, простота використання та налаштування – все використання і опрацювання відбувається в автоматичному режимі із трансляцією на карту, розташовану на сервері у реальному часі, налаштування може бути виконано за лічені хвилини використовуючи розроблену схему. Запропонована система позбавлена проблем відомих аналогів пов'язаних зі специфічністю обробки інформації та довготривалістю роботи.

На рис. 8 зображено з'єднання мікроконтролера на платі STM32F4DISCOVERY із Wi-Fi модулем ESP8266. З'єднання необхідно здійснити згідно до табл. 1.

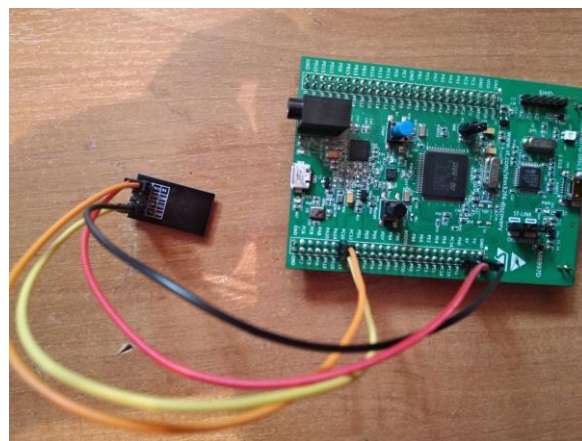


Рис. 8. З'єднання мікроконтролера на платі STM32F4DISCOVERY із Wi-Fi модулем ESP8266

Персонально до кожної локальної системи необхідно задати параметри у програмі. Необхідно встановити швидкість роботи мікроконтролера на інтерфейсі зв'язку із Wi-Fi модулем.

ТАБЛИЦЯ 1. ТАБЛИЦЯ З'ЄДНАНЬ

Вихід ESP	Вихід STM32
ESP RX	TX PC10 – USART3_TX
ESP TX	RX PC11 – USART3_RX
ESP 3V3	3V
ESP GND	GND

Після встановлення параметрів увімкнення модуля – необхідно доповнити код програми, а саме встановити дані точки доступу для підключення – логін, пароль, а також MAC адреса точки доступу. Для серверної частини ніяких змін не потрібно, можна використовувати все, як із коробки, лише необхідно у самій програмі задати IP адресу самого серверу для прослуховування. Теж саме стосується і додатків для системи та геопозиції, лише у мікромодуля, що відповідає за GOOGLE MAP потрібно ввести власний GOOGLE API KEY. Не обов'язково мати, чи вводити ключ. У випадку відсутності ключа – карта все одно буде створена, але с деякими обмеженнями.

ДОДАТКИ

[1] McEwen A., Cassimally H. «Designing the Internet of Things» – K.: John Wiley and Sons, Ltd., 2014. – 337 с.

[2] Сталлингс Уильям «Интернет вещей: сетевая архитектура и архитектура безопасности»: «Информационный сборник», 2017

[3] «STM32 vs Arduino» [Електронний ресурс] - <https://habr.com/ru/post/191054/>

[4] «Обзор микроконтроллеров семейства STM32F4» [Електронний ресурс] - <http://fpga.in.ua/dsp/dsp-theory/obzor-mikrokontrollov-semejstva-stm32f4.html>

[5] STMicroelectronics, "Discovery kit with STM32F407VG MCU," May 2017. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00039084-discovery-

kit-with-stm32f407vg-mcu-stmicroelectronics.pdf. [Accessed 25 July 2020].

[6] «Подключение Wi-Fi модуля ESP8266 к микроконтроллеру STM32 (STM32L051K6T6)» [Электронный ресурс] - <https://wiredlogic.io/ru/diy-wi-fi-esp8266-esp12-stm32-open-source-ru/>

[7] «The Internet of Things: Network and Security Architecture», The Internet Protocol Journal Vol 18, No 4, 2015

[8] «The Future of Smart Cities and Sensors» - «Advanced MP Technology», 2016

[9] «How Smart Sensor Systems Will Revolutionise the Future» - «Appcessories», 2016

[10] «Как сделать беспроводную Wi-Fi-сеть: сражаемся с цифровой агорафобией» [Электронный ресурс] - <https://www.it-world.ru/tech/admin/144399.html>

[11] «Стандарты, протоколы и технологии связи для подключения IoT-устройств» [Электронный ресурс] - <https://iot.ru/promyshlennost/standarty-protokoly-i-tehnologii-svyazi-dlya-podklyucheniya-iot-ustroyst>