

ТЕЛЕКОМУНІКАЦІЇ

УДК 681.518.5

АНАЛІЗ ІНІЦІАТИВИ АЛЬЯНСУ ZIGBEE ЩОДО СТВОРЕННЯ УНІВЕРСАЛЬНОЇ МОВИ DOTDOT ДЛЯ ІОТ

БЕЗРУК В.М., ВЛАСОВА В.О., КРИВЕНКО С.А.

Розглядається будова ZigBee для створення мережі моніторингу та контролю за станом ліній електропередачі. Технології IoT традиційно використовують в мережах з низькою швидкістю, низькою потужністю і короткими пакетами даних. Аналіз зосереджений на прикладі реалізації концепції "розумного міста" з підтримкою додатків IoT. Експериментальні результати показують ефективність і стабільність роботи системи.

Ключові слова: Інтернет речей, перетворення зірки в сітку, томографія електричного опору, ZigBee.

Keywords: Internet of Things, star-mesh transform, Electrical resistivity tomography, ZigBee.

Вступ

Багато комунікаційних технологій добре відомі, такі як Wi-Fi, Bluetooth, ZigBee і стільникові 2G/3G/4G. Але є кілька нещодавно створених нових варіантів мереж, наприклад, Thread як альтернатива для додатків домашньої автоматизації, і технологій ширококомовного телебачення, що реалізуються у великих містах для більш широкого сегмента на основі використання IoT. Залежно від застосування такі фактори, як діапазон частот, вимоги до даних, вимоги безпеки та потужності і терміну служби батареї диктують вибір тієї чи іншої форми поєднання технологій.

В січні 2017 року некомерційний альянс ZigBee Alliance представив dotdot як універсальну мову для Інтернету речей IoT [1].

Група говорить, що мова dotdot приймає IoT на рівні додатків ZigBee і дозволяє йому працювати в різних мережних технологіях. Це важливо, оскільки в даний час більшість IoT пристроїв не говорять однією мовою, навіть якщо вони використовують ту саму бездротову технологію. Результатом є Інтернет речей, в якому є клаптеві переклади, що зроблені в хмарі. Розробники платформи і додатків повинні підтримувати зростаючий набір унікальних інтерфейсів для кожного постачальника продуктів. Альянс ZigBee був заснований в 2002 році і в даний час налічує понад 400 членів. Альянс створив малопотужну бездротову технологію сітчастої (MESH) мережі для підключення. А в 2007 році він випустив відкриту мову, яка підтримує сотні різноманітних інтелектуальних пристроїв у будинках і на підприємствах. Альянс ZigBee вже працює на dotdot з ще однією некомерційною групою Thread Group. Мережі цієї групи на базі Інтернет протоколу з'єднують інтелектуальні домашні пристрої. Бездротова сітчаста мережа ZigBee, мережа IP

Thread засновані на стандарті IEEE 802.15.4. Але це не є обов'язковою вимогою для роботи dotdot. Чи є мова dotdot необхідною?

Джахангир Мохаммед, генеральний менеджер групи IoT компанії Cisco Jasper, сказав: «Якщо у вас є 100 пристроїв, ви, ймовірно, маєте 99 операційних систем. Це явна фрагментація». «Мова dotdot ZigBee, безумовно, може допомогти з цією фрагментацією», пише Робб Хеншо, керівник глобальних комунікацій для IoT Хмари в Cisco Jasper, в електронному листі до SDxCentral. «Ще існує необхідність спільної мови, яка дозволить всім пристроям спілкуватися, з допомогою dotdot мови спільнота ZigBee вирішує правильні проблеми». Але є й інші конкуруючі групи, такі як AllSeen Альянс і Weave Google екосистеми, які намагаються вирішити ту саму проблему. «З плином часу промисловість, ймовірно, побачить, як одна з цих груп або мов виникне як переможець» говорить Henshaw. Cisco Jasper любить віддалитися від таких груп, як ZigBee, які він вважає орієнтованими на споживчі товари, на відміну від корпоративного класу і промислового устаткування. «Оголошення Dotdot насправді не має ніякого впливу на Cisco Jasper», говорить Хеншо. «Наша платформа IoT являє собою автоматизовану платформу управління зв'язком, а мова додатку не має значення для нас, тому що ми агностики в цьому сенсі». Альянс ZigBee зробив собі ім'я в споживчих товарах IoT, до складу групи дійсно входять такі компанії, як Huawei і Schneider Electric, які роблять промислові IoT [2]. Автоматизація деяких сфер підвищує безпеку життєдіяльності людини, а безпека людини зараз є пріоритетним напрямком у розвитку технологій.

Ідеї створення інтелектуальних систем, які б допомагали забезпечувати людині комфорт і безпеку, були запропоновані вже досить давно, проте в той час реалізувати ці ідеї було неможливо через відсутність ресурсів і технологій, що дозволяють це зробити.

Вираз "Розумне місто", останнім часом став дуже популярним. При цьому настільки широкий і, по суті, невизначений, що його використовують в самих різних контекстах і з самими різними цілями. Існує безліч ідей по створенню таких міст у світі. Пропонують найрізноманітніші концепції "розумного міста" і його побудови. В даній статті розглянуто аспект контролю за станом ліній електроживлення, що є одним з найважливіших факторів забезпечення якості життя.

Метою дослідження є аналіз можливостей альтернативних підходів на прикладі застосування мови Python як універсальної для різних операційних систем IoT.

1. Модель IoT

1.1. Модель ZigBee

Мова dotdot є універсальною мовою Інтернету речей, що дає можливість для смарт-об'єктів виконувати спільну роботу в будь-якій мережі. Аналогічні можливості має мова Python, яка дозволяє проводити для смарт-об'єктів спільну роботу в будь-якій мережі на основі інфраструктури веб-клієнтів та веб-серверів.

Типове використання одного з кластерів ZigBee наведено на рис. 1 [3].

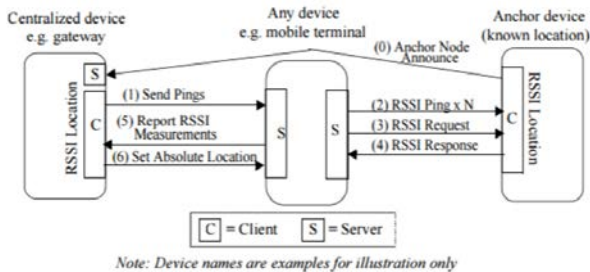


Рис. 1. Типове використання одного з кластерів ZigBee

Протоколи ZigBee розроблені для використання у вбудованих додатках, що вимагають низьку швидкість передачі даних і низьке енергоспоживання. Мета ZigBee – це створення недорогої мережі, що сама зорганізується, з сітчастою топологією, призначеної для вирішення широкого кола завдань. ZigBee призначений для мобільних пристроїв, де необхідна тривала робота від батареї, тому ключовим завданням є зменшення часу роботи передавачів. Цей час визначається пропускну здатністю каналу при заданому об'ємі інформації, яку необхідно передати. Як приклад можна розглянути передачу інформації, якщо реалізуються дві топології – зірка та сітка для п'яти вузлів (рис. 2).

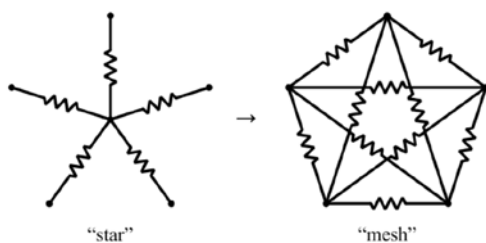


Рис. 2. Перетворення топології мережі

Для порівняння цих двох підходів необхідні кількісні характеристики, які розглядаються в наступному підрозділі.

1.2. Передача даних у IoT

У чистому вигляді, при ефірній передачі, швидкість передачі даних становить 250 Кбіт/с для кожного каналу в діапазоні 2.4 ГГц, 50 кбіт/с для кожного каналу в діапазоні 915 МГц. Відстань передачі – від 10 до 75 метрів та понад 1500 метрів

для ZigBee Pro, хоча вона дуже залежить від параметрів конкретного обладнання і середовища передачі. Максимальна вихідна потужність радіовипромінювання в основному становить від 0 до 20 дБм (від 1 до 100 мВт). Стандарт передбачає пропорційне збільшення потужності передавача для збільшення швидкості передачі даних [4].

Можна виділити 3 основних способи взаємодії з інтернет-речами:

- прямий доступ;
- доступ через шлюз;
- доступ через сервер.

У разі прямого доступу інтернет-речі повинні мати власну IP-адресу або мережевий псевдонім, за яким до них можна звернутися з будь-якого клієнтського додатку. Інтерфейс з такими речами зазвичай виконаний у вигляді веб-ресурсу з графічним інтерфейсом для управління за допомогою веб-браузера. Можливе використання спеціалізованого програмного забезпечення.

Недоліки такого способу очевидні:

- необхідність мати фіксовану адресу в мережі, що залежить від провайдера послуги зв'язку з Інтернетом таких речей. Іншим виходом із ситуації є використання аліаса (мережевого псевдоніма IP-адреси), що вимагає постійного звернення інтернет-речі до спеціального сервера з запитом про оновлення мережевої адреси за псевдонімом;

- ліміт підключень до пристрою – викликано низькою якістю зв'язку інтернет-речей, а також їх слабкими обчислювальними ресурсами. Така проблема вирішується шляхом включення до складу інтернет-речі високопродуктивного обладнання та підключення речей до стабільного джерела зв'язку з Інтернетом. Це викликає необхідність у більшому споживанні енергії і часто змушує робити такі речі стаціонарними, що живляться від постійних джерел електроенергії.

Доступ до інтернет-речей через шлюз є більш раціональним способом організації взаємодії і повністю витісняє метод прямого доступу в разі необхідності організації зв'язку бездротових сенсорних мереж або мережі інтернет-речей з глобальною мережею Інтернет. Більшість стандартів бездротових сенсорних мереж не підтримують протокол IP, використовуючи власні протоколи взаємодії. Така особливість викликає необхідність наявності пристрою для ретрансляції повідомлень з сенсорної мережі в мережу Інтернет для сумісності протоколів.

Недоліки такого підходу ті ж, що й у випадку прямого доступу, але поширюються вони на шлюз.

Доступ до інтернет-речей через сервер має на увазі наявність посередника між інтернет-речами

і користувачем. Таким посередником є сервер, в основні функції якого входить:

- прийом повідомлень від інтернет-речей і передача їх користувачам;
- зберігання прийнятої інформації та її обробка;
- забезпечення користувацького інтерфейсу з можливістю двостороннього обміну між користувачем і інтернет-реччю.

Такий спосіб доступу є найбільш раціональним і часто використовуваним, оскільки дозволяє перенести навантаження обробки запитів користувачів з інтернет-речей на централізований сервер, тим самим розвантажуючи слабкий радіо-канал зв'язку інтернет-речей, перерозподіляючи навантаження на дротові канали зв'язку між сервером і користувачами.

Метод централізованого сервера також надає надійні засоби зберігання і обробки інформації, дозволяє інтернет-речам взаємодіяти один з одним і користуватися хмарними обчисленнями. Даний підхід може використовувати метод шлюзу для з'єднання локальних бездротових мереж з сервером.

У наведеному проекті використовується метод локального сервера, яким виступає Raspberry Pi 3.

1.3. Взаємодія елементів

Більшість інтернет-речей є самостійними пристроями, що передають деяку інформацію в базу даних для подальшої обробки та передачі кінцевим користувачам. Для збору, зберігання, обробки, спрощення пошуку, контролю та візуалізації інформації використовується локальний сервер управління інтернет-речами, що включає в себе:

- модуль обробки інформації;
- базу даних для зберігання інформації, що збирається;
- інтерфейс взаємодії з інтернет-речами (деякий протокол, що підтримується всіма речами);
- систему контролю користувача доступу до речей, управління їх ієрархією, параметрами і функціоналом.

Останній пункт може бути перенесений на окремий локальний чи централізований сервер і винесений на відокремлену систему управління, відділяючи один від одного базу даних, що збираються, і базу даних управління речами.

Модуль обробки інформації може проводити розпакування стислої для прискорення транспортування інформації, проводити арифметичні і логічні обчислення, класифікацію і перетворення прийнятої інформації до зрозумілого формату сприйняття.

Для зберігання інформації з підведенням деякої статистики, необхідної для моніторингу об'єктів, а також надання актуальної опублікованої інформації всім користувачам незалежно від їх часу підключення використовується база даних.

Управління інтернет-речами проводиться також через сервер. Користувач відправляє команду на сервер управління, а він, у свою чергу, транслює її адресату. Набір команд управління повинен бути мінімальним і загальним для всіх інтернет-речей для забезпечення повного відділення управління від даних. Це важлива вимога до інтернет-речей, оскільки дозволяє значно спростити обмін інформацією між користувачами і речами, а також зробити крок на шляху до стандартизації методів взаємодії в Інтернеті речей і створення загального центру управління.

Відділення керуючої інформації від даних передбачає виділення мінімального набору команд і сутностей, над якими ці команди можуть бути виконані. Наприклад, для вимикання лампочки, включення тостера або перемикання режиму зйомки відеокамери можна виконувати одну і ту ж команду – запис. Даними в такому випадку будуть стани пристроїв. Для отримання інформації необхідно виконати зчитування поточного стану пристрою або окремої його складової частини.

2. Результати дослідження

Архітектура Інтернету речей передбачає наявність таких функціональних рівнів: мережа датчиків, шлюз, управління, додаток. Оскільки нижній рівень складається з датчиків, сенсорів та актуаторів, то відразу ж виникає необхідність в специфічних протоколах для забезпечення взаємодії цих пристроїв один з одним і верхніми рівнями. Стандартні прикладні протоколи не підходять через їх непристосованість до умов мережі Інтернету речей. Датчик, зазвичай мініатюрний, з невеликою пам'яттю, вимірює фізичні параметри в режимі реального часу, найчастіше в умовах низького енергозабезпечення. Результати вимірювань обробляються сенсорним вузлом і передаються на сервер. Обсяг інформації, що формується одним сенсорним вузлом, порівняно невеликий, проте більшість сервісів Інтернету речей побудовано на принципі обробки інформації від безлічі вузлів, що принципово відрізняється від архітектури, прийнятої в класичних мережах, типу «абонент – вузол зв'язку» для телефонії, «клієнт-сервер» – для передачі даних.

IoT була застосована до пристроїв для поліпшення їх характеристик на прикладі забезпечення перетворення топології. Була застосована підписка на сервіс Event Hub Azure IoT для польового експерименту. Різні нові додатки обробки даних програмного забезпечення за допомогою фреймворка (Python Web Framework) і SQLight перебувають на стадії розробки для подальшого підвищення продуктивності мережі і ефективності обслуговування великої кількості пристроїв IoT.

2.1. Датчик-потенціометр

Нижче викладені основні концепції перетворення трьох опорних Microsoft IoT проектів: індикатора, що блимає; IoT веб браузер; датчик-потенціометр [6]. В запропонованому проекті був реалізований потенціал мікросхеми MCP3008, яка має вісім несиметричних входів. Наші інтелектуальні результати можуть бути розглянуті на прикладі переходу від одного входу до двох входів.

Два поворотних потенціометри і два світлодіоди LED підключені до аналого-цифрового перетворювача та Raspberry Pi 3. Аналого-цифровий перетворювач на основі протоколу SPI був використаний для читання величини резисторів і для управління LED на основі положення щіток потенціометрів.

Були використані такі компоненти: два світлодіоди; два резистори 330 Ом; один резистор 10 кОм; один 10-бітний аналого-цифровий перетворювач MCP3008; два 10 кОм тримерних потенціометри; Raspberry Pi 3 IoT; один макет і кілька дровтів; Raspberry Pi 7" сенсорний рідкокристалічний екран.

Нижче (рис. 3) наведена інформація щодо мікросхеми MCP3008.

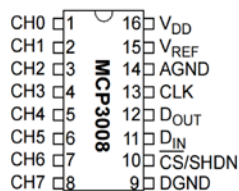


Рис. 3. Аналого-цифровий перетворювач MCP3008

Мікросхема MCP3008 є 10-бітний аналого-цифровий перетворювач з вісьмома несиметричними входами. Він здатний формувати до 200 кілобайт в секунду.

Нижче (рис. 4) наведена інформація щодо RaspberryPi3.

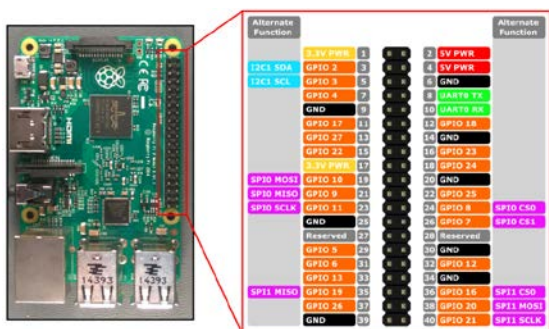


Рис. 4. RaspberryPi3

Схема (рис. 5) була зібрана в такий спосіб.

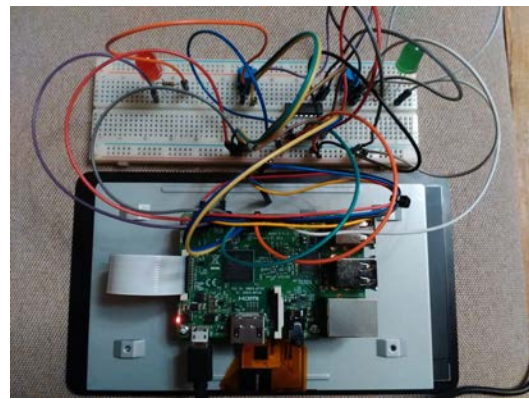


Рис. 5. Схема з'єднання елементів IoT

Мікросхема MCP3008 була підключена так:
MCP3008: VDD – 3.3V на Raspberry Pi 3;
MCP3008: VREF – 3.3V на Raspberry Pi 3;
MCP3008: AGND – GND на Raspberry Pi 3;
MCP3008: CLK – «SPI0 SCLK» на Raspberry Pi 3;
MCP3008: Dout – «SPI0 MISO» на Raspberry Pi 3;
MCP3008: Din – «SPI0 MOSI» на Raspberry Pi 3;
MCP3008: CS/SHDN – «SPI0 CS0» на RaspberryPi 3;

MCP3008: DGND – GND на Raspberry Pi 3;
MCP3008: CH0 – повзунок нульового потенціометра;

MCP3008: CH1 – повзунок першого потенціометра.

Червоний, зелений і синій світлодіоди були підключені до контактів з 4 до 6. Відповідно змінні були записані у файл MainPage.xaml.cs рішення Microsoft Visual Studio.

2.2. Запуск запропонованого проекту IoT

Три часткові проекти: індикатор, що блимає; IoT веб-браузер; і потенціометр були об'єднані в єдиний файл проекту (головна сторінка C#) рішення Microsoft Visual Studio 2015. Мікросхеми MCP3008 і ARM для цільової архітектури були обрані в запропонованому проекті.

Мінливі цифри на екрані показують напругу, що змінюється.

Величина напруги пропорційна величині опору в цій схемі. Загалом ця напруга може бути використана в різних ZigBee пристроях.

Коли напруга більша, ніж половина діапазону аналого-цифрового перетворювача, світлодіод включений. Діапазон дорівнює 1024 для мікросхеми MCP3008. В іншому випадку, світлодіод гасне.

2.3. “Багато-до-багатьох” інтерфейс

Багато-до-багатьох інтерфейс має дві сторони: сторону сервера або шлюзу та сторону додатків або інфраструктури. IoT пристрій було підключено на стороні додатків, Microsoft Azure був підключений на стороні сервера. В пристрої IoT застосовано рішення, що складається з трьох проектів: управління LED; IoT браузер; потенціометричний датчик. Датчик вимірює вісім опорів у

зіркоподібній топології. Події збираються на додатку Hub, потік подій аналітично обробляється відповідно налаштованим екземпляром IoT Hub на Microsoft Azure. Браузер IoT обробляє запит датчиків та повертає відповідь на сторону сервера, використовуючи функцію зворотного виклику, коли вона була надана (рис. 6).



Рис. 6. Відображення результатів виміру

Висновки

Аналіз можливостей альтернативного підходу на прикладі застосування мови Python як універсальної для різних операційних систем IoT показав таке. Експериментальні результати свідчать про ефективність і стабільність роботи системи із застосуванням IoT на стороні додатків та дата-центру обробки даних Microsoft Azure – на стороні сервера.

Література:

- [1] The zigbee alliance: Foundation and Future of the IoT [Online] Available: <http://www.zigbee.org/the-zigbee-alliance-foundation-and-future-of-the-iot-by-john-e-osborne-ii-chairman-of-the-board/>
- [2] Zigbee Writes a Universal Language for IoT [Online] Available: <https://www.sdxcentral.com/articles/news/zigbee-writes-universal-language-iot/2017/01/>

[3] STANDARDS: ZIGBEE CLUSTER LIBRARY [Online] Available:

<http://www.zigbee.org/download/standards-zigbee-cluster-library/>

[4] IEEE Standard for Low-Rate Wireless Networks. Sponsored by the LAN/MAN Standards Committee. IEEE Std 802.15.4™-2015. P. 510 [Online] Available: <http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf>

[5] Calabrese G. Symmetrical components applied to electric power networks. New York, Ronald Press co., 1959, 31-32.

[6] MS IoT samples [Online]. Available: <https://github.com/ms-iot/samples/archive/develop.zip>

Надійшла до редколегії 12.04.2017

Рецензент: д-р техн. наук, проф. Бараннік В.В.

Безрук Валерій Михайлович, д-р техн. наук, академік академії зв'язку України, професор, завідуючий кафедрою «Інформаційно-мережева інженерія» ХНУРЕ. Адреса: Україна, 61166, Харків, пр. Науки, 14. E-mail: valeriy_bezruk@ukr.net

Власова Вікторія Олександрівна, канд. техн. наук, старший викладач кафедри «Інформаційно-мережева інженерія» ХНУРЕ. Адреса: Україна, 61166, Харків, пр. Науки, 14. E-mail: zlata_ne@bk.ru

Кривенко Станіслав Анатолійович, канд. техн. наук, доцент кафедри «Інформаційно-мережева інженерія» ХНУРЕ. Україна, 61166, Харків, пр. Науки, 14. E-mail: Stanislav.Kryvenko@nure.ua

Bezruk Valeriy, Doctor of Technical Sciences, professor, academician of the Ukrainian Academy of Telecommunications, Professor, Head of the Department of Information and network engineering, Kharkiv National University of Radio Electronics. Address: 14 Nauki ave, Kharkiv, Ukraine, 61166. E-mail: valeriy_bezruk@ukr.net

Vlasova Viktoriya, Candidate of Technical Sciences, Senior Lecturer Department of Information and network engineering, Kharkiv National University of Radio Electronics. Address: 14 Nauki ave., Kharkiv, Ukraine, 61166. E-mail: zlata_ne@bk.ru

Krivenko Stanislav, Candidate of technical sciences, associate professor of the Information and network-engineering department, Kharkov National University of Radioelectronics. Address: 14 Nauki ave, Kharkiv, Ukraine, 61166. E-mail: Stanislav.Kryvenko@nure.ua