

РОЗРОБКА МОДЕЛІ ЕЛЕМЕНТІВ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ КОНТРОЛЮ ПАРАМЕТРІВ НАВКОЛИШНЬОГО СЕРЕДОВИЩА

В статті розглядається питання створення активного датчику мережі «Інтернет речей». Інтернет речей представляє собою активну мережу, що складається з простих датчиків, вузлів підключення до мережі Інтернет та серверної частини. Програмне забезпечення датчиків створюється з використанням високих мов програмування, таких як C/C++. Простота створення ПО датчика дозволяє легко імплементувати його в 8-бітні контролери за необхідності досягнення простоти керування. Легкість доступу до WiFi надає можливість швидкого розгортання мережі.

Ключові слова: датчик, сервер, Інтернет речей.

N.I. PRAVORSKA
Khmelnitskyi National University

DEVELOPMENT OF MODEL OF ELEMENTS OF THE INTERNET OF THINGS IS FOR CONTROL OF PARAMETERS OF ENVIRONMENT

In the article the question of creation active to the sensor of network is examined the Internet of things. The internet of things is an active network that consists of simple sensors, connecting knots to the network the internet and server side. Main purposes of IoT sensors are gathering different kind of information. Sensors do this accurately by requests from server or by interval.

Article deal with IoT sensor of humidity and temperature. Software of sensors is created with the use of high as if programming, such as C/C++. Simplicity of creation sensor' software allows easily implement it into 8-bit controller. This allows achieve necessity level of management simplicity. Lightness of access to Wi-Fi gives possibility of rapid network development.

Keywords: sensor, server, internet of things.

Вступ

Інтернет речей (Internet of Things, IoT) – це міжмережева взаємодія фізичних пристроїв, транспортних засобів (що також називаються "підключеними пристроями" або "розумними пристроями"), будівель й інших предметів, забезпечених електронікою, програмним забезпеченням, датчиками, актуаторами і мережевим підключенням, які дозволяють цим об'єктам збирати і обмінюватися даними. У 2013 році Світова ініціатива стандартизації Інтернету речей (Global Standards Initiative on Internet of Things, IoT – GSI) визначила Інтернет речей як "інфраструктуру інформаційного суспільства". Інтернет речей дозволяє пристроям бути виявленим або контрольованим видалено через існуючу мережеву інфраструктуру, створюючи можливості для прямої інтеграції фізичного світу в комп'ютерну систему, в результаті підвищуючи ефективність, точність, економічну вигоду на додаток до скорочення втручання людини. Коли Інтернет речей доповнюється датчиками і виконавчими механізмами, ця технологія стає базою для загальніших кібер-фізических систем, які також охоплюють такі технології, як розумні мережі, розумні будинки, розумний транспорт і розумні міста. Кожна річ унікально ідентифікується через вбудовану обчислювальну систему, але здатна взаємодіяти з існуючою інфраструктурою інтернету. За оцінками експертів, до 2020 року в Інтернет речей буде включено майже 50 мільярдів об'єктів [1].

Сам термін "Інтернет речей" був придуманий Пітером Т. Люїсом у 1985 р. під час сесії Федеральної комісії із зв'язку США (FCC), що підтримує безпроводний зв'язок на 15-й Законодавчій конференції конгресу "Чорний конгрес". У своїй промові він заявив, що "Інтернет речей – це інтеграція людей, процесів і технологій з пристроями, що підключаються, і датчиками для забезпечення видаленого моніторингу, стану, маніпулювання і оцінки тенденцій таких пристроїв".

Основна частина

Згідно із звітом компанії Gartner Inc. до 2020 року Інтернет речей складатиметься з приблизно 20.8 мільярдів пристроїв. Інтеграція з мережею Інтернет має на увазі, що пристрої повинні мати унікальні ідентифікатори, якими є IP-адреса. Проте, у зв'язку з обмеженістю пулу адрес IPv4 (4.3 мільярдів унікальних адрес), об'єкти в мережі «Інтернет речей» повинні використати IPv6 для розміщення надзвичайно величезного адресного простору. Ці об'єкти представляють не лише пристрої з сенсорними можливостями, але і активації (замки, контрольовані через інтернет), що забезпечують можливість. Для подальшого росту майбутнє Інтернету речей не представляється можливим без підтримки IPv6; і, отже, глобальне впровадження IPv6 найближчими роками матиме вирішальне значення для успішного розвитку Інтернету речей в майбутньому.

Окрім домашнього управління енергією, Інтернет речей особливо актуальний для SmartGrid, оскільки він надає системи збору і обробки інформації про енергію і потужність в автоматичному режимі з метою підвищення ефективності, надійності, стійкості і економічності розподілу і виробництва електроенергії. Використання облаштувань розширеної вимірювальної інфраструктури (AMI), підключених до інтернету, облаштування електропостачання можуть не лише збирати інформацію від кінцевих споживачів, але і управляти іншими облаштуваннями автоматизації розподілу, такими як трансформаторами і АПВ (автоматичне повторне включення).

Існують декілька запланованих або таких, що проводяться нині широкомасштабних розгортань Інтернету речей, щоб забезпечити краще управління містом і міськими системами. Наприклад, місто Сонгдо, що в Південній Кореї, перший у своєму роді повністю обладнане і підключене в єдину мережу розумне місто, яке практично збудовано. Практично кожен елемент цього міста планується підключити, з'єднати і перетворити на постійний потік даних, які спостерігатимуться і аналізуватимуться безліччю комп'ютерів, з нульовим або невеликим втручанням людини.



Рис. 1. Розумне місто (Smart City)

Мережева архітектура

Інтернет речей вимагає величезної масштабованості в мережевому просторі для обробки швидкозростаючої кількості пристроїв. Стандарт 6LoWPAN використовуватиметься для підключення пристроїв до IP-мереж. З мільярдами пристроїв, що додаються в мережевий простір, IPv6 відіграватиме важливу роль в обробці масштабованості мережевого рівня. Протокол обмежених застосувань, MQTT і ZeroMQ забезпечать легке перенесення даних. Туманні обчислення (Fog computing) є реальною альтернативою запобіганню такому великому потоку даних, що надходить через інтернет. Обчислювальна потужність граничних мережевих пристроїв (маршрутизатори, комутатори) можуть використовуватися для аналізу і обробки даних, що забезпечує простоту масштабування в режимі реального часу.

Реалізація концепції "розумного датчику"

Для вирішення проблеми впровадження IoT можуть бути використані різні способи. Але оптимальним рішенням є устаткування житла датчиками "розумного" будинку. Для ефективної роботи, згідно з рис. 1, потрібні лише декілька типів датчиків: датчик рівня води, датчик вологості, освітлення, вимірювання інших величин. Для організації спільної роботи набору датчиків потрібний контролер, який збиратиме і оброблятиме інформацію, а також пересилати її, припустимо, мобільний телефон користувача для інформування про проблему. Можливе додаткове підключення до системи кульових електронів або насосів.

У якості такого контролера використовуються будь-які процесорні засоби, наприклад 8-, 16- та 32-бітні контролери. Обсяг пам'яті, обсяг даних визначається типом даних. Розглянемо реалізацію найбільш простого варіанту елемента IoT – датчика вологості та температури. У якості доступу до мережі використаємо найбільш популярний спосіб – доступ до бездротової мережі WiFi. Для створення прототипів таких датчиків широке застосування знайшли мікроконтролери фірми «Microchip» серії ATMEGA в результаті злиття з компанією «Atmel».

Для швидкого створення проектів керування використовується середовище Arduino. Arduino – це проект з відкритим початковим кодом, компанія, що створює апаратне і програмне забезпечення. Найчастіше, мікроконтролери програмуються з використанням функцій мов C і C++. На додаток до використання традиційних інструментальних засобів компіляції, проект Arduino надає інтегроване середовище розробки (IDE), засноване на мові Processing [6]. Програми, написані в IDE для Arduino, називаються нарисами або скетчами. Arduino IDE підтримує мови C і C++, використовуючи спеціальні правила структуризації. У Arduino IDE вбудована бібліотека з проекту Wiring, що надає доступ до безлічі простих процедур вводу/виводу. Код, написаний користувачем, може містити всього 2 базових функції, для запуску нарису і основного циклу програми, які компілюються і зв'язуються з функцією-заглушкою main() у виконувану циклічну програму за допомогою GNU toolchain, також включеною в дистрибутив IDE.

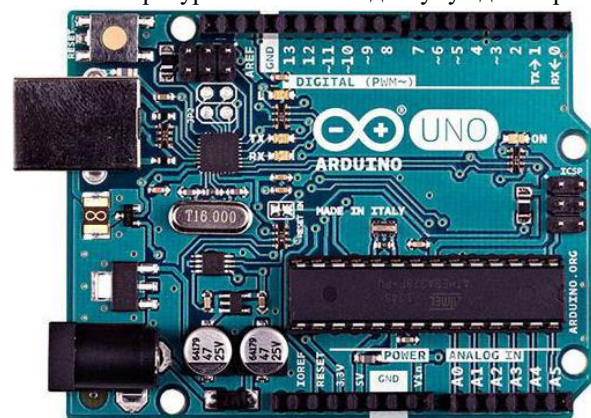


Рис. 2. Контролер Arduino Uno R3

Програмна частина

Створення програмного коду визначає необхідність взаємодії 2-х частин:

- 1) серверної – для обробки, зберігання та прийняття рішень щодо взаємодії з виконавчими вузлами;
- 2) клієнтської – для накопичення, передачі на сервер та виконання отриманих з сервера команд.

Розглянемо створення такого коду для елементарного датчику температури та вологості з доступом в інтернет.

Структурна схема показана на рис. 3.



Рис. 3. Структурна схема датчика погоди

У якості датчика температури використано датчик DHT11, у якості модуля доступу до мережі – ESP8266. Розглянемо найбільш типові функції, що використовуються для зчитування

```

#include "DHT.h"
#define DHTPIN 2 // Контакт підключення модуля DHT11 до контролера
DHT dht(DHTPIN, DHT11);
float D_t, D_h;
void Weather_Setup() {
    dht.begin();
}
void Weather_Get() {
    D_h = dht.readHumidity(); // вологість
    D_t = dht.readTemperature(); // температура
}
  
```

Рис. 4. Фрагмент коду ініціалізації датчика та коду зчитування параметрів

На рис. 4. показано простий код, що дозволяє занести значення вологості та температури у пам'ять контролера. Частота викликів може бути встановлена в програмі контролера, наприклад, зчитування кожні 5 хвилин або кожні 10 секунд стану навколо датчику.

Модуль ESP8266 забезпечує підключення до мережі з використанням простого послідовного інтерфейсу керування. На рис. 5 показано фрагмент коду щодо встановлення такого з'єднання.

```

#include <SoftwareSerial.h>
SoftwareSerial wifiSerial(2, 3); // RX, TX сигнали

void setup() {
    wifiSerial.begin(9600);
    wifiSerial.setTimeout(5000);
    Serial.begin(9600);
    wifiCmd("AT+RST", 1000, "Ready"); //повідомлення про готовність

    boolean connected = false; //підключення до мережі
    for(int i=0; i<5; i++) {
        if(connectWiFi()) { // перевірка підключення із заданими параметрами
            connected = true;
            break;
        }
    }
    while (!connected);
    delay(5000);
}
  
```

Рис. 5. Встановлення з'єднання з модулем ESP8266

Підключення до довільного вузла WiFi мережі реалізовано за допомогою командами AT запиту "AT+CWJAP=" (рис. 6).

```

boolean connectWiFi() {
    wifiCmd("AT+CWJAP=" , 1000, "OK");
    wifiCmd("AT+CWJAP=\"Network\", \"Password\"", 2000, "OK");
    return true;
}
  
```

Рис. 6. Підключення до довільного вузла WiFi мережі

І довершується підключення вже запитом до відповідного серверу у форматі GET запиту (рис. 7). На сервері має бути підготовлено відповідний файл скрипта, що написано на одній з мов взаємодії (рис. 8).

```
//устанавливаем режим соединения SINGLE
wifiCmd("AT+CIPMUX=0", 1000, "OK");
}

boolean wifiRqst() {
url = "/weather.php?Temp="+ (String)D_t+"&Press="+ (String)D_h();
WiFi.begin("name", "pass");
Serial.print("GET "+url+" HTTP/1.1");
Serial.print("Host: <a href='http://server.ua' >server.ua</a>");
Serial.print("User-Agent: WeatherWiFi/1.1");
Serial.print("Connection: close");
Serial.print("\n");
}
```

Рис. 7. Передача значень GET запитом

```
<?PHP
include ('config.php');
$db = @mysql_connect($db_loc,$db_user,$db_pass);
@mysql_select_db($db_name,$db);
mysql_query("SET TIME_ZONE='+3:00'");
$a=$_GET["Temp"];
$b=$_GET["Press"];
mysql_query("INSERT INTO bmp280 (Temp,Press) VALUES ('$a','$b')");
mysql_close($db);
?>
```

Рис. 8. Файл запису отриманих значень на стороні сервера ("weather.php")

Висновки

Швидкість впровадження, одночасно з масовістю, доступністю складових частин, а також зручність IDE для створення програмного коду дозволяють досягти зручності в виконанні задач розвитку вузлів Інтернету речей.

Література

1. Internet of Things. URL: https://en.wikipedia.org/wiki/Internet_of_things (дата обращения 16.03.2017).
2. Privacy of the Internet of Things. URL: <https://arxiv.org/ftp/arxiv/papers/1611/1611.03340.pdf> (дата обращения 22.04.2017)
3. Arduino. URL: <https://en.wikipedia.org/wiki/Arduino> (дата обращения 15.03.2017).
4. Современные сетевые технологии, технологии Интернет / Нижегородский государственный университет им. Н.И. Лобачевского. URL: <http://www.itlab.unn.ru/Uploads/part2.pdf> (дата обращения 15.03.2017).
5. Корнелл Гари, Кей С. Хорстманн. Core Java Volume II: Advanced Features, 2004. 1088 с.

References

1. Internet of Things. URL: https://en.wikipedia.org/wiki/Internet_of_things (data obrasheniya 16.03.2017).
2. Privacy of the Internet of Things. URL: <https://arxiv.org/ftp/arxiv/papers/1611/1611.03340.pdf> (data obrasheniya 22.04.2017)
3. Arduino. URL: <https://en.wikipedia.org/wiki/Arduino> (data obrasheniya 15.03.2017).
4. Sovremennye setevye tehnologii, tehnologii Internet / Nizhegorodskij gosudarstvennyj universitet im. N.I. Lobachevskogo. URL: <http://www.itlab.unn.ru/Uploads/part2.pdf> (data obrasheniya 15.03.2017).
5. Kornell Gari, Kej S. Horstmann. Core Java Volume II: Advanced Features, 2004. 1088 s.

Рецензія/Peer review : 6.9.2019 р. Надрукована/Printed : 02.01.2020
Стаття рецензована редакційною колегією