

ПОДХОД ЗА АВТОМАТИЧЕН ЗАПИС И ВИЗУАЛИЗАЦИЯ НА ДАННИ ОТ АВТОНОМНИ СИСТЕМИ

Делян Генков

Технически университет - Габрово

AN APPROACH FOR AUTOMATIC RECORDING AND VISUALIZATION OF DATA FROM AUTONOMOUS SYSTEMS

Delyan Genkov

Technical University - Gabrovo

Abstract

Internet of Things becomes very popular technology nowadays, not only in home systems, but also in many industrial fields of application. The technology presumes many autonomous systems in form of controllers with sensors attached to measure some data, and to send the measured values for centralized storage and management. There are many possible approaches for this technology – including cloud computing, fog computing and combined approaches.

This paper presents one possible approach for automatic sending, recording and visualization of data from many devices to a central location.

Keywords: Internet of Things, data, centralized management, network.

ВЪВЕДЕНИЕ

Технологията Интернет на нещата (Internet of Things) представлява свързване на сензори и/или изпълнителни механизми към евтини контролери с ограничени ресурси и възможности, които могат да се свързват към компютърна мрежа/Интернет за да бъдат наблюдавани данните, следени от устройството и подавани команди към изпълнителните механизми ръчно от оператор или автоматично чрез вградена логика в софтуера на контролера. Тя е в основата на някои съвременни тенденции за домашна, сградна и промишлена автоматизация и изграждането на т. нар. „умен дом“ (Smart Home), „умен град“ (Smart City) и др. Между 2011 и 2015 г. в Барселона е изградена IoT мрежа, управляваща „цифрови автобусни спирки“, паркинги, градско осветление, напояване на градските паркове и много други. Подобна концепция се използва и в доста индустриални решения, например при наблюдението и управлението на газопроводи, електро и водопреносни мрежи,

влакови трасета, производствени поточни линии и други.

Развитието и използването на технологията поставя проблемите за изпращането на данните през компютърната мрежа и/или Интернет, събирането им на централно място (сървър, облачна среда), където те могат да бъдат визуализирани, обработвани и съхранявани.

Най-често се използва един от следните три модела или комбинация от тях:

- Клиент-сървър - този модел е добре познат в Интернет. На сървъра работи програма, която при извикване от страна на клиента извършва определени действия – показва информация или предизвиква действие. Моделът е подходящ за малък брой контролери. При IoT решенията се използват главно две реализации. При първата сървър се намира в контролера (най-често web сървър), към който клиентът през локалната мрежа или Интернет се свързва с програма (най-

често web браузър) и наблюдава и/или управлява параметрите на системата. При втората сървърът се намира някъде в мрежата (най-често локалната), а към него се свързват контролерите и предават своите параметри. Клиентът също се свързва към сървъра, наблюдава записаните параметри и може да издава команди към сървъра, които той препраща към контролерите за изпълнение.

- Cloud Computing (облачна среда) – много IoT решения идват с готова облачна услуга – собственикът регистрира акаунт в облачната система на производителя, където регистрира своите устройства и получава интерфейс, чрез който може да наблюдава и управлява контролерите си. За целта разбира се е необходимо и контролерът и клиентът да имат достъп до Интернет. В облачната услуга могат да се комбинират множество контролери, намиращи се на различни места.
- Fog Computing - този модел е подходящ за комбинирането на голям брой сензори, при него в контролера се прави предварителна обработка на данните, приемани от неговите сензори и към централното устройство (сървър или облак) се предава само обобщена информация с цел да се избегне затрупването му с огромен обем данни.

В настоящата разработка е представен клиент-сървър модел за автоматичен запис на данни, изпратени от множество контролери в база данни, възможност за визуализация, обработка на данните и реализация на команди, които да бъдат изпращани обратно към контролера с цел управление при настъпване на предварително дефинирани условия, например включване на охлаждащ компонент при достигане на праг на измерваната температура.

ИЗЛОЖЕНИЕ

За реализация на задачата за автоматичен запис и визуализация на данни от авто-

номни системи е използвана бесплатната програмна система с отворен код - ThingsBoard. [2]

Системата предлага възможност за инсталация под управлението на операционни системи Windows и Linux, инсталация върху контролер Raspberry PI 3, в облачна среда – Google Cloud или Amazon AWS EC2 или в контейнерна реализация, използвайки Docker. В настоящата разработка е използвана инсталация под Microsoft Windows 10, но една от основните цели на работата е да се използва инсталация в облачна среда.

Необходимите условия за стартиране на системата са минимум 2 GB оперативна памет, като е препоръчително поне 4 GB и инсталирана Oracle Java 8. Данните могат да се записват във вградената в системата HSQLDB база данни, което е приложимо за тестови цели. За сериозни приложения се препоръчва използването на външна система за управление на база данни, както препоръчаните от производителя са PostgreSQL или NoSQL база данни Casandra.

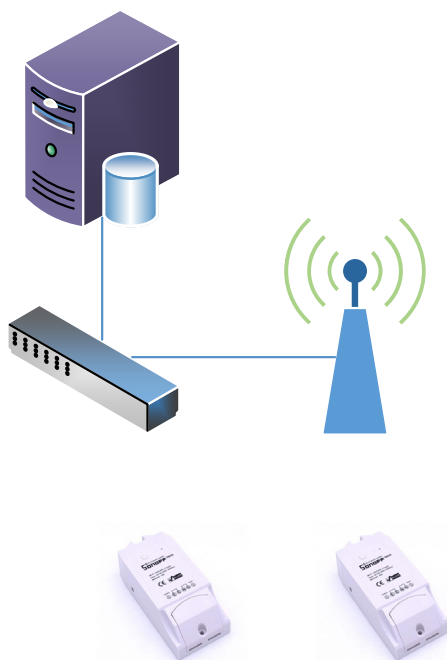
Към софтуерната система може да се свърже произволен контролер, който може да изпраща необходимите данни чрез JSON заявки. В разработката са използвани контролери на фирма SONOFF, които имат процесор ESP8266 и вградена безжична свързаност.

Един от използваните контролери е представен на фигура 1.



Фиг. 1. Контролер SONOFF TH10

На фигура 2 е показана постановката на системата.



Фиг. 2. Структура на системата

Използвани са следните компоненти: ролята на сървър изпълнява компютър с процесор Intel Pentium G2120 Dual Core CPU @ 3100 MHz, 8GB RAM, 1 TB HDD, Windows 10 и инсталирана програмна система ThingsBoard с база данни PostgreSQL и Java 8. Мрежовият комутатор е TP-Link TL-SG3210. Ролята на безжична точка за достъп се изпълнява от маршрутизатор TP-Link TL-WR1043ND, към който са свързани по безжичен път два контролера. Единият е SONOFF TH10, към който е свързан сензор за измерване на температура и влажност на въздуха DHT22. Вторият контролер е SONOFF POW, който има вграден сензор за консумирана мощност и захранващо напрежение.

Програмирането на контролерите се извършва с безплатния инструмент Arduino IDE v.1.8.7, където е необходимо да се изтеглят и инсталират следните библиотеки:

- ESP8266 core for Arduino версия 2.4.2 за базовите функции на контролера;
- PubSubClient версия 2.6.0 за комуникацията със сървъра;

Необходимите настройки, които трябва да се направят в програмата на контролера са:

```
//Задаване на SSID на безжичната мрежа
#define WIFI_AP "....."
//Задаване на паролата за мрежата
#define WIFI_PASSWORD "....."
//Парола на конкретното устройство за
//достъп до сървъра
#define TOKEN "SONOFF_POW_55"
//Цифров вход на контролера, на който е
//свързан сензора
#define DHTPIN 14
//Тип на сензора за температура и
//влажност
#define DHTTYPE DHT22
//IP адрес на сървъра
char thingsboardServer[] = "192.168.1.12";
//TCP порт, на който работи сървъра
int serverPort = 1883;
```

Заявката, която изпраща устройството към сървъра е във формат JSON и описва параметрите, които се измерват, заедно с текущите стойности на измерването. Примерна заявка за описание на температура и влажност е показана на фигура 3.

```
{
  "temperature": 20,
  "humidity": 76
}
```

Фиг. 3. Формат на JSON заявка

В текущата разработка данните се предават всяка секунда, но честотата на заявките може да зависи от броя контролери и от спецификата на измерваната величина, както и от необходимото време за реакция на системата.

```
WiFiClient wifiClient;
DHT dht(DHTPIN, DHTTYPE);
PubSubClient client(wifiClient);
int status = WL_IDLE_STATUS;
unsigned long lastSend;
```

Основните редове от програмния код на контролера за показани в следващия фрагмент. Както всяка програма за Arduino и тази съдържа две основни функции – setup(), която се изпълнява при началното включване на системата и loop(), чието изпълнение се повтаря непрекъснато до изключване на захранването на контролера.

```

void setup()
{
  dht.begin();
  delay(10);
  InitWiFi();
  client.setServer(thingsboardServer,
1883);
  lastSend = 0;
}

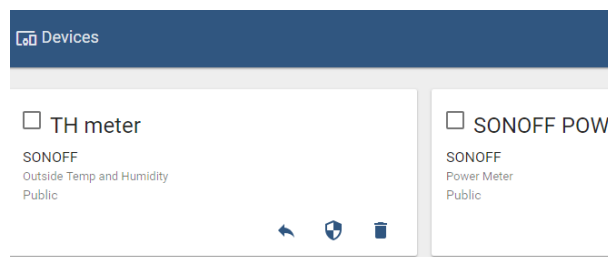
void loop()
{
  if ( !client.connected() ) {
    reconnect();
  }
  if ( millis() - lastSend > 1000 ) {
    getAndSendData();
    lastSend = millis();
  }
  client.loop();
}

void getAndSendData()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    return;
  }
  String payload = "{";
  payload += "\"temperature\":";
  payload += temperature; payload +=
  ",";
  payload += "\"humidity\":"; payload
  += humidity;
  payload += "}";
  char attributes[100];
  payload.toCharArray(  attributes,
100 );
  client.publish(
"v1/devices/me/telemetry",  attributes
);
}

```

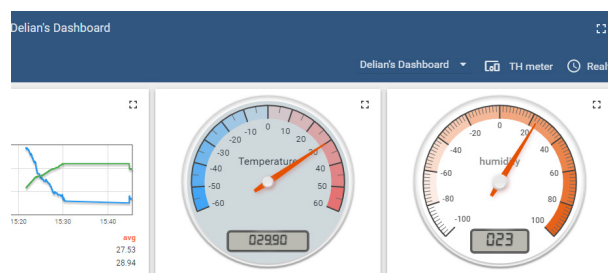
За работата на постановката е необходимо да се настои и софтуера Thingsboard. След първоначалните задачи - регистрирането на потребителско име, парола и други необходими параметри в системата е необходима регистрацията на всеки контролер. Възможните методи за автентикация за Token (парола, която се записва в конфигурацията) или X.509 сертификат, ако използваме активна директория. В текущата разработка се използва token (парола).

На фигура 4 е показана регистрацията на устройство в системата.



Фиг. 4. Регистрация на устройства

Резултат от работата на системата е показан на фигура 4.



Фиг. 4. Резултати

ЗАКЛЮЧЕНИЕ

Изградената система е тествана с два контролера, измерващи различни величини. Предстои тестването на постановка с повече контролери и сензори. Предвижда се използването на платформата да се мигрира към облачни технологии, което да допринесе за по-широкото приложение на подхода за различни ситуации.

БЛАГОДАРНОСТИ

Настоящият документ е изготвен с финансовата помощ на договор № 1810E за провеждане на научни изследвания по проект на тема: „Приложение на виртуализационни и облачни технологии в обучението“ към Технически университет – Габрово.

REFERENCES

- [1] Ajunament de Barcelona, Barcelona Digital City, <https://ajunament.barcelona.cat/digital/en>, дата на използване: 04.10.2018 г.
- [2] ThingsBoard, Open-source IoT Platform, <https://thingsboard.io/>, дата на използване 04.10.2018 г.