BLE통신을 이용한 IoT 스마트홈 모니터링 시스템 개발

즈엉꽁딴1), 김명균2)

Development of an IoT Smart Home System Using BLE

Duong Cong Tan 1), Myung Kyun Kim 2)

요 약

최근 사물인터넷 (IoT) 기술이 많은 응용 분야에 적용되어 단말장치 및 응용들이 폭발적으로 증가하고 있다. IoT 시스템은 센서와 액츄에이터들이 연결되어 상호 메시지를 교환하면서 필요한 응용을 수행하는 시스템이다. 본 논문에서는 홈 디바이스들을 쉽게 모니터링 및 제어할 수 있는 IoT 시스템을 개발하였다. 본 논문의 시스템은 센서단말장치, IoT게이트웨이, IoT서버로 구성되어 있으며, 센서단말장치는 주변을 센싱하여 수집한 센서데이터를 BLE 통신을 이용하여 IoT게이트웨이를 통해 서버로 전송하며, Rfduino를 이용하여 구현하였다. 센서단말장치와 서비 사이의 통신을 중개하는 IoT게이트웨이는 Raspberry Pi를 이용하여 구현하였고, IoT서버는 삼성 ARTIK cloud를 이용하여 구현하였다. 구성한 시스템은 대학 실험실내에 설치하여 테스트를 수행하였고, 실험결과 설치 및 관리가 용이하며 추가로 다른 기능을 쉽게 확장할 수 있음을 보여주었다. 또한 ARTIK cloud의 rule 정의, action 메시지 기능을 통해 홈 디바이스에 대한 파라미터 제어 기능을 용이하게 구현하였다.

핵심어: 사물인터넷, 아틱 플랫폼, Rduino, 스마트홈

Abstract

Recently, the Internet of Thing (IoT) technology is expanding explosively in the number of devices and applications in many application areas. In IoT systems, sensors and actuators are connected to the Internet and cooperate to do some applications by exchanging information. In this paper an IoT smart home system is developed to monitor and control home devices easily. Our system consists of sensor devices, IoT gateways, and an IoT server. Sensor devices developed using Rduino sense the physical world and transmit their data using BLE to the IoT server thru the IoT gateway. We implemented the IoT gateway using Raspberry Pi and the IoT server using ARTIK cloud. We installed our system and made a test in our lab, which showed that our system can be installed and managed easily and extended its functionality in an easy way. By taking advantage of the rule mechanism and action messages of ARTIK cloud, we implemented the control of device parameters easily by sending action messages to the ARTIK cloud.
Keywords : Internet of Things (IoT), ARTIK platform, RFduino, Smart Home

1. Introduction

Internet of Things (IoT) is the next stage of Internet’s evolution which involves interlinking of various uniquely identified physical and virtual things and devices[11]. Managing of these IoT devices is more complex due to the heterogeneity of the system and limited availability of the resources. Many research challenges are identified and it is commonly accepted that the IoT paradigm is still in their infancy[1,2]. The heterogeneity and the scalability are considered as important requirements for IoT. In fact, the existing IoT architectures[12] are basically relied on a hierarchical structure to support these requirements. However, almost of them are fixed number of end devices or difficult to upgrade system when have more demand to add more end device into the system. Moreover, exist cloud for IoT do not allow user manages the behavior to control end device automatically. In helping to resolve this challenge, in this paper, we propose an IoT system for smart home using RFduino module supporting Bluetooth Low Energy (BLE) [3] and ARTIK Cloud a new IoT Cloud designed by Samsung. This IoT system allow user and developer easier to develop and add more end device join into system. Besides, they allow user can manage the behavior to control end device base taking advantage of ARTIK Cloud feature.

RFduino module [4] is small multi-controller board which is studded with huge numbers of features to makes it the most efficient device. It uses CPU 16MHz 32-bit ARM Cortex-M0 with 128KB Flash memory and 8KB for RAM. Moreover, RFduino is peripheral BLE device, enables to communicate with other BLE device. ARTIK cloud [5] is an open data exchange platform for the IoTs, designed to accelerate device interoperability and enable new data insights. Using ARTIK cloud, clients can use many different protocols to send and receive data, including REST, WebSockets, MQTT and CoAP. A special feature of ARTIK cloud make it different with other, this is Rules. It allows to define the Rules that send Action to control end devices when triggered by incoming messages.

The remainder of this paper is organized as follows. Some backgrounds are given in Section 2. The design system is presented in Section 3. Section 4 describes experimental results. The conclusion is given in Section 5.

2. Background

2.1 Artik Cloud

ARTIK Cloud is an open data exchange platform for the IoTs, designed to accelerate device
interoperability and enable new data insights by Samsung [5]. It is designed to be as easy as possible to integrate with existing devices and services. ARTIK Cloud is device-agnostic and scales to all IoT applications, including smart homes, smart cities, and industrial IoT. Developer can think beyond a single device and focus on making new connection around data. They call this Data Driven Development. The First, clients can access and aggregate historical data from different sources, thus opening a new perspective on big data. The second, clients can use many different protocols to send and receive data, including REST, WebSockets, MQTT and CoAP. The last, ARTIK Cloud is the only service that gives users complete control over their data. By granting access to devices and applications, users promote an ecosystem of services around data. In order to do so, the ARTIK Cloud present the basic concept behind how ARTIK Cloud recognizes devices, data and applications.

Data is stored in ARTIK Cloud by applications and devices in a message. ARTIK Cloud recognizes two message types: one carries only data and other carries actions for devices. Each message in ARTIK Cloud is associated with a set of identifying metadata: the device ID, user ID and application ID. To record a message, an application must provide the payload and the device ID. ARTIK cloud can automatically infer the user ID and application ID from the access token that was generated during authentication. Applications must specify at least one of these parameters when querying ARTIK Cloud. This means that depending on the scenario, data can be queried per user, per device or per application.

### 2.2 Bluetooth Low Energy (BLE)

Bluetooth Low Energy is a low power, short range wireless communication technology. This technology, also known as Smart Bluetooth, was introduced as a part of Bluetooth Core Specification version 4.0 and was adopted in 2010 [6,7]. The major purpose of this technology is to enable transceivers with lower power consumption, low complexity and lower cost than the ones possible with the classic Bluetooth technology. The latest release of Bluetooth adds new capabilities in Bluetooth LE that make it a suitable technology for low-power devices in the Internet of Things[9,10].
BLE is a connection oriented wireless technology, so two devices which want to exchange data must enter a fixed connection before a data transmission is possible. The Fig. 1 will present the state of BLE device during operating time. In the state “Advertising” BLE devices (so called advertisers) can transmit advertisement data like a broadcaster to scanning devices. Furthermore advertising BLE devices can signalize scanning and initiating devices that they are ready to go into a fixed connection. Only three of 40 data channels are used for the advertising communication. A device in the state “Scanning” (a scanner) and the state “Initiating” will listen for packets from a specific advertising device. This device will than respond to these packets to initiate a new connection. A device in the state “Connection” is called being in a connection. There are two different roles within a state “Connection”. These roles are named as “Master Role” and “Slave Role”. When changing from the state “Initiating” to the state “Connection” the device will have the Master Role. When entering the state “Connection” from the state “Advertising” the device will fulfill the Slave Role. Important to know the Master defines all timings of the transmission.

3. System Design

The IoT system we are proposing is depicted in Fig. 1. There are four components, is comprised of end devices, gateway, IoT Cloud and user application. End devices can be included many kinds of sensors or actuators in order to monitor and make the appropriate response based on activities of actuators. In the gateway, it provides common function for interconnecting and works as a translator to exchange data properly between end devices and IoT cloud. To do so, it need to support many kinds of communication as Zigbee, BLE, WiFi for end devices and CoAP, MQTT, WebSocket for IoT cloud. Moreover, gateway stores and manages the information of end devices including name, mac address, device id and device token. This information will be stores at the first time when an end device joins
into the system. The important component in system is IoT cloud, which has capabilities to register, store and manage the collection data from end devices. Finally, an application provides for user to display useful information and make interface to allow user can control equipment in the system.

In our system implementation, we use RFduino module as end devices, Raspberry PI3 as gateway and ARTIK cloud as IoT server. Because we use only BLE communication between end devices and Gateway and a gateway have multiple BLE connection to end devices, so end devices have a role as slave and gateway as master.

3.1 Virtual Device on Artik Cloud

Each ARTIK Cloud device corresponds to a device type which will stores and handle data follow respective actual devices. A device type has a unique Manifest that describes the data sent by the device and any commands it can receive. So, before create a device on ARTIK Cloud, we have to define what type of data that actual end device will send to ARTIK Cloud. This will determines the data fields that are included in the Manifest. Similar with data, we have to define what actions will be controlled on actual end device. These actions are also describes in the Manifest. For example, the actual end device include a simple smart light. It will send state of light to ARTIK cloud and it have two action: setOn and setOff. We will have to create a instance of device type which have the manifest as Fig. 3

Note that creating a virtual device have to do first before implement application for end devices. When we complete creating a virtual device on ARTIK Cloud, we can obtain DEVICE_ID and DEVICE_TOKEN and we use them to performs register with gateway. The implement of end devices in
this system have to base on mechanism, which are describe in next part.

3.2 Processing step of non-registered and registered devices

In this paper, we use BLE to make connection between end devices and gateway. Thus, end devices and gateway have to perform initial process before they actually exchange data. The initial process allow new end devices registered to join into the system and handle every case which can happen during operating time and create a prototype which developer can easily follow to develop and install new device for system. So, we define the processing for Non-registered devices and registered devices.

The detail sequence when a new device join into the IoT system is depicted in the Fig. 4. After powering up, end devices will broadcast advertisement message (ADV.MSG) over the air if it is in disconnected state. The ADV.MSG includes access key which is to determine end device for the system or not. The gateway will perform scanning regularly during working time to overhear joining request from end device. Once overhear a ADV.MSG (1), gateway will check access key in ADV.MSG to make decision to send connection request (2) if matching. After connected, gateway will send request information message to end devices (3) and waiting receive information device (4) including name, mac address, DEVICE_ID, DEVICE_TOKEN and store them into configuration file. Gateway use this information to make connection (6) (7) with ARTIK Cloud and send ready message (8) to device if the connection with cloud is established successfully. From this time, end device goes to communication mode and can exchange data with system.

For the registered devices, the mechanism is similar with mechanism for non-registered device in step (1) and (2), But after connected, gateway will check mac address of end devices compare with
information in configuration file to determine that device is registered or not. In this case, the devices already registered, so gateway get the information in configuration file to register connection to cloud and then send ready message (5) to device. After that, it goes to communication mode and can exchange data with system. Fig. 5 shows the processing step for the registered devices.

During working time, sometimes the connection between device and gateway is broken by some reason such as out of battery or moving far away out of transmission range, the device will go to disconnected state and gateway will close connection to cloud of that device. Then end device will broadcast ADV_MSG to try join into system again if possible.

4. System Implementation

In our system, we use three end devices using Rfduino module as shown in Fig. 6. The first end device is to monitor temperature, brightness and detect people in the room, the second one is to detect air-conditioner status and control air-conditioner, and the third one is to detect light status and control the light.

Gateway in our system is run on Raspberry Pi 3, which is run on raspbian OS – a light distribution of Linux. Our implementation uses node.js language chooses MQTT and WebSocket protocol to develop with using noble[8] library for BLE communication program.

On the ARTIK Cloud, we create four virtual devices, three of them are for three corresponding actual end devices to store respective data and actions and a virtual device is for user application. The system will support two mode: normal mode and auto control mode. The normal mode is every action in system will be sent by user. And auto control mode is every action will be performed automatically on ARTIK cloud based on the collected data in cloud.
With auto control mode, we define Rules to send action based on some condition. For example, when we enter into the room, the motion sensor detect has people and it check the brightness value, if it is less than a threshold the action turns on light will be sent to end devices to turn on the light. All the tests run both normal mode and auto control mode work well. After powering on the gateway and end devices, end devices take a few seconds to perform register a new device join into the system and then they can exchange data. The Fig. 7 is console log for installing and exchange data of new device process at gateway. The system can self-control and handle well when we change to auto control mode. The action will be performed near real-time, it takes from user send action to actually perform at end device. And Fig. 8 shows the screen of the smart-phone application.

5. Conclusions

In this paper, we design a system to monitor and control IoT environment. Defining a prototype and mechanism to allow developer easily implement and upgrade system is key element of proposed system. Also we show that using ARTIK Cloud is a plus to allow user can change behavior of server when they are in auto control mode. The experimental results also evaluated performance and advantage of the system.
References


