

Implementation of Smart Farm Devices using Open Source Software

Sunghyun Yoon, Mun-Hwan Choi, Juyoung Park

Protocol Engineering Center, Electronics and Telecommunications Research Institute

218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, KOREA

shy72@etri.re.kr, mhchoi@etri.re.kr, jypark@etri.re.kr

Abstract—With the recent development of IoT technology, a variety of products and services that were previously not implemented due to technical limitations are gradually becoming feasible. Accordingly, in the agricultural field, smart farm implementation is gradually spreading through the use of IoT technology. In this paper, we analyse standard technologies for smart farms through implementation of smart farm devices using IoTivity, an open source software provided by OCF which is a middleware standard technology for logical interoperability between IoT devices.

Keywords—Smart Farm, Smart Agriculture, IoT, OCF, IoTivity

I. INTRODUCTION

With the Internet of Things (IoT) attracting attention as the core of the next-generation ICT industry, and attempts to apply it to various fields are expanding, one of the areas that is receiving the most attention is smart farm. Smart farm is a cutting-edge agricultural technology that seeks to create high added value by converging IoT with existing agriculture to increase productivity and efficiency [1].

The smart farm consists of sensor devices for monitoring growth of crops and environmental information (e.g. temperature, humidity, CO₂, soil, nutrients, etc.), actuators that controls the smart farm facilities (e.g. window switch, ventilation, heating, irrigation, watering, etc.), controllers that control devices attached to each facility, and an operation management system that manages them in an integrated manner [2].

Smart farm equipment currently being distributed is not guaranteed to be compatible with each other because product specifications are different for each manufacturer. This problem is an obstacle for farmers to adopt a variety of equipment. In other words, even if they want to introduce new equipment from other manufacturers with specific functions, they cannot do it because it is not compatible with the existing equipment. In addition, manufacturers have fewer sales opportunities, as compatibility is not guaranteed unless all equipment is supplied at once. Therefore, securing the interchangeability of smart farm equipment becomes a very important requirement for both farmers and manufacturers.

The OCF (Open Connectivity Foundation) [3] is an international standardization organization related to IoT, and many IoT companies around the world are participating in it, and it has a comparative advantage over other standardization

organizations in scale and ecosystem expansion by simultaneously developing open source projects.

The OCF supports interoperability between various IoT devices by using various wired/wireless connection technologies. Therefore, many IoT-based services such as smart home, automobile, logistics, and healthcare can be developed through the OCF. One of the biggest advantages of the OCF is the free patent policy, which allows the OCF members to use all patents related to the OCF technologies free of charge.

OCF operates in a client-server model based on the resource model. A device that provides resources operates as a server, and a device that requests resources operates as a client. As an example applied to a smart farm environment, smart farm equipment such as sensors and actuators is a server, and an operation management system capable of controlling smart farm facilities through a user interface may be equipped with a function as a client. This means that even home appliances such as TVs, including smartphones and tablets, can be smart farm equipment as clients depending on the compatibility between all devices that support the OCF. Therefore, when OCF is applied to a smart farm, smart farm devices that were not compatible in the past can be connected to each other, making a significant contribution to the productivity improvement of farmers.

In this paper, we analyse OCF based smart farm data model standard technology, and implement a test bed using IoTivity [4], an open source software provided by OCF. Through this, a concept verification for the application of standard technologies to smart farms that can guarantee mutual compatibility between IoT devices is performed.

II. OCF & IOTIVITY

OCF is a global standards organization that develops an IoT open platform. OCF precedes standardization in connection with IoTivity open source. IoTivity is an open source project sponsored by OCF and an IoT open source technology based on OCF standards, which implements all OCF mandatory requirement.

OCF follows the 3-pillar alignment policy as the basic principle of standardization. This is a policy in which standards can be established and revised only when standard document preparation, open source code implementation, and

certification process development are all preceded, thereby ensuring compatibility with standards and interoperability between products.

A. OCF Overview

The beginning of OCF is the OIC (Open Interconnect Consortium) organization. OIC is an Internet of Things standard organization established by four companies of Samsung Electronics, Intel, Cisco, and MediaTek as founding board members in September 2014, and had been active until February 2016. In December 2015, OIC integrated and merged with the UPnP Forum which was a representative international standard organization for smart home, and has grown to more than 100 member companies. Currently, the UPnP Forum maintains its name by working as an UPnP Working Group under OCF. The growth of OIC continued even after UPnP was merged, and in February, three members of the AllSeen Alliance, another IoT open source organization, Microsoft, Qualcomm, and Electrolux participated in OIC as new board members. At this time, the name of the organization was changed from OIC to OCF.

OCF adopts a very original operating method that simultaneously develops open source and standard specifications. In other words, OCF is playing the role of developing open source and spreading it to the outside at the same time as developing standard specifications. This aims to reduce the burden that each company has to develop its own software after the development of the specification, while allowing it to be immediately installed and spread in the market at the same time as the OCF standard development [5].

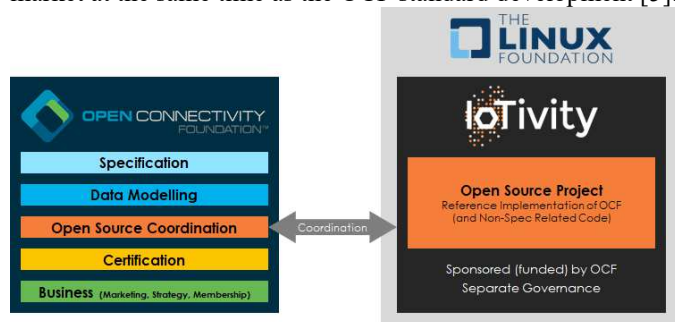


Figure 1. OCF open source and standards development method

In particular, as shown in Figure 1, OCF operates OCF open source through a non-profit organization (Linux Foundation) dedicated to operating open source projects to accelerate the spread of external developer participation in open source development, and the project name is IoTivity.

It has communicated with open source developers, and is developing software jointly. IoTivity is developed based on the OCF standard, and is being developed while accommodating various IoT technologies, and OCF supports the operating costs required for IoTivity development.

In other words, from the perspective of external open source developers, IoTivity is a pure open source project that has nothing to do with OCF, and products that adopt IoTivity can naturally comply with the OCF standard to secure interoperability between products. In addition, OCF is a

membership standard organization, so membership registration is required to participate in OCF, but IoTivity is a pure open source project, so any developer who wants to use it without signing up as an OCF member can use it, modify, redistribute, and commercialize it.

In particular, OCF is operating a program to certify products developed in compliance with the standard, and by giving the official logo of the OCF to the products that passed the inspection necessary for certification through a plugfest between member products.

B. IoTivity concept and composition

1) IoTivity Concept

The purpose of IoTivity is to develop an open source software framework that can seamlessly connect billions of devices that will appear in the future IoT world, regardless of various operating systems and network protocols.

One of the most important parts for revitalizing the IoT ecosystem is how easily can small and medium-sized businesses that manufacture various things add Internet connection functions to their products and provide an environment in which smartphone apps or services can be provided together. IoTivity is a framework for satisfying these requirements, ensuring interoperability between IoT devices and enabling rapid IoT product development. In addition, it is continuously expanding and applying the latest IoT technologies that are newly developed, and is expanding the range of support for various open source hardware (e.g. Arduino, Raspberry Pi, Edison, etc.) and software platforms (e.g. Android, Linux, Windows, etc.). In addition, IoTivity is basically an OIC standard-based IoT middleware open source technology that supports interoperability.

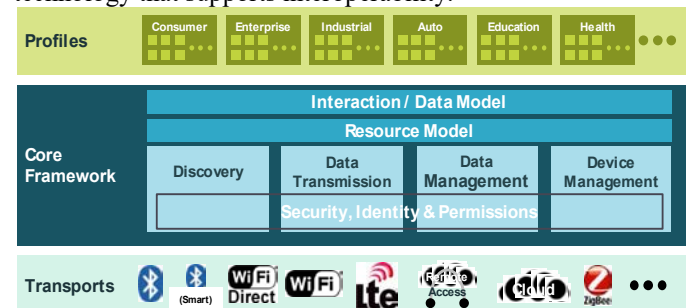


Figure 2. Conceptual Architecture of IoTivity

Figure 2 shows the conceptual architecture of IoTivity. The IoTivity is composed three layers including transports layer, core framework layer and profiles layer.

The Transports layer is a part of various connection technologies used in IoT such as Bluetooth, Wi-Fi, and Zigbee. The profile layer means each vertical field of IoT applications such as consumers, companies, automobiles, and health, and are developed independently of the connection technology of the transport layer. The core framework layer supports core functions for interoperability such as resource discovery, data transmission, device management, and data management.

In the case of the transport layer, new technologies can be continuously expanded, and even if such expansion is performed, applications of the profile layer can be executed without change with the support of the framework layer.

For reference, the licensing policy follows Apache 2.0 and is operated by the Linux Foundation.

2) IoTivity Composition

IoTivity is based on a resource-based RESTful architecture model, thus expressing all things as resources and providing CRUDN (Create, Read, Update, Delete and Notify) operations. In addition, it is designed based on CoAP (Constrained Application Protocol) without a daemon, so it is easy to support low-spec and low-power devices [6].

The detailed composition of the IoTivity framework is shown in Figure 3.

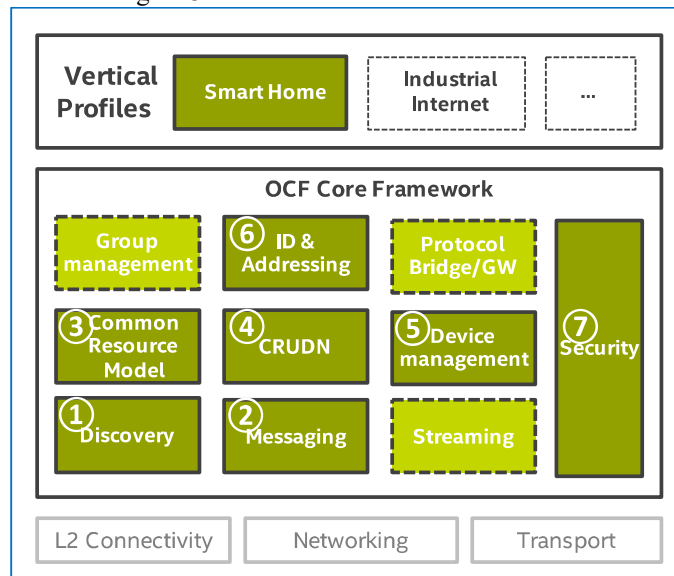


Figure 3. IoTivity framework

As shown in Figure 3, the IoTivity core framework a number of functions for interoperability between IoT devices such as resource discovery, messaging, general resource model, CRUDN operations, device management, identifier and address expression, security and etc. IoTivity's essential messaging protocol is IETF CoAP, and it also supports HTTP (HyperText Transfer Protocol) and MQTT (MQ Telemetry Transport) as optional messaging protocols in the future.

III. IMPLEMENTATION OF SMART FARM DEVICES

A. Smart Farm Overview

Smart farm is a cutting-edge agricultural technology that seeks to create high added value by converging ICT such as IoT technologies with traditional agriculture to increase productivity and efficiency, and various IT devices are used. Depending on the application field, smart farms can be classified into smart greenhouses for growing crops, smart livestock for livestock, and smart orchards for growing fruit trees. That is, smart farms can be classified according to the

growing target, and further subdivided according to the breed of the growing target. As such, various types of facilities and devices may be provided in the smart farm according to the type and method of farming, and the case of a smart greenhouse is described as an example. The following figure is an example of a typical smart greenhouse.

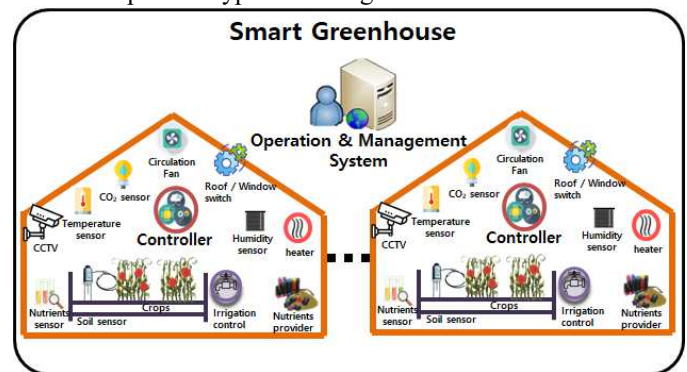


Figure 4. Typical smart greenhouse configuration example

As shown in Figure 4, each smart greenhouse is equipped with sensors and actuators, and a controller that receives environmental and equipment status data from them, and transmits control commands for them.

There are various types of sensors such as Temperature, Humidity, Carbon Dioxide (CO2), Soil, and Nutrient Sensors, and each sensor delivers the current measurement value at a fixed period or upon request.

Likewise, there are many types of actuators for environmental control of smart greenhouses such as roof/window switch, circulation fan, heating, irrigation control, nutrient provider, and imaging device (CCTV). They perform their own operation according to the control signal from controller.

The controller transmits a signal for controlling the actuator device, and also transmits the data transmitted by the sensor and the actuator to the operation management system.

The operation management system integrates and manages a plurality of greenhouses, and is used like an operating system that monitors and controls the conditions of each greenhouse operated by the farm.

The following figure is a typical example of the composition of smart greenhouse equipment.

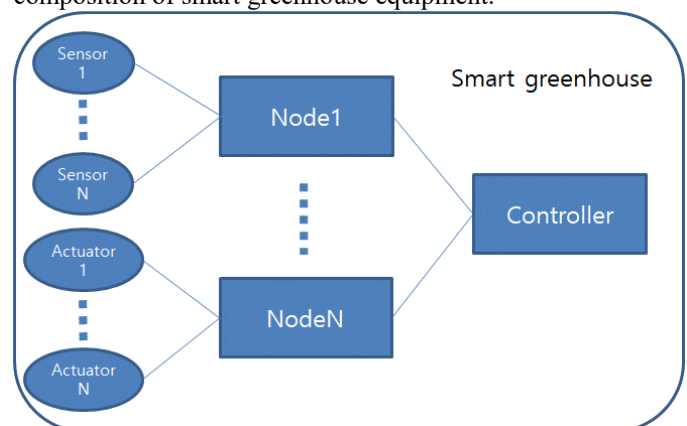


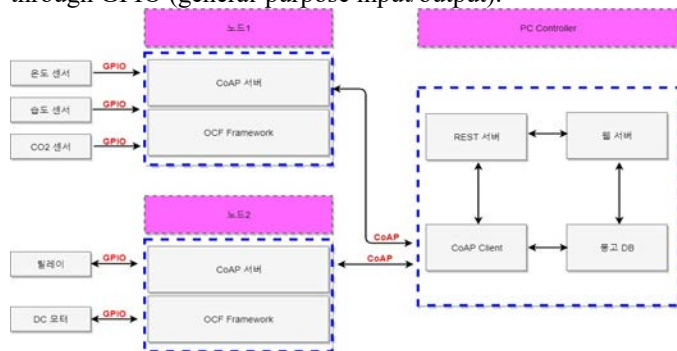
Figure 5. Typical configuration example of smart greenhouse equipment

Equipment such as a sensor and a driver generally operates by connecting a plurality of sensor and driver devices to a sensor and driver node. Usually, several sensor modules are connected to the sensor node, and several actuator modules are connected to the driver node. In the case of a composite node in which sensor modules and actuator modules are connected to one node together, it has not yet been widely spread for reasons such as cost and management. Each node may be connected to a separate controller for various reasons such as different manufacturers or special purpose nodes, but is usually connected to one controller.

B. Smart Farm Testbed

Figure 6 is the configuration of the OCF-based smart farm testbed implemented in this paper. The smart farm testbed consists of nodes that plays the role of sensors and actuators (ocf-server), and a controller that plays the role of controller (ocf-client).

The node that plays the role of a server providing resources consists of a sensor node connected to sensor modules such as temperature, humidity, and CO₂, and an actuator node connected to actuator modules such as relay and DC motor. The sensor node and actuator node are implemented in the Raspberry Pi, and the sensor and actuator modules operate through GPIO (general-purpose input/output).

**Figure 6.** Smart farm testbed configuration

After booting, the sensor node periodically transmits the status to the DB server. The sensor node immediately responds to the current sensing value when a message requesting the status is received from the web server (user UI). After booting, the actuator node transfers the status to the DB server periodically or irregularly. When the actuator node receives the control message, it performs a predetermined operation. Message delivery is delivered through CoAP.

The server node is equipped with an IoTivity-based CoAP server to periodically transmit resource data generated by the sensor module to the CoAP client.

The control node that plays the role of a client corresponding to the controller that receives data from the server node and transmits control messages to the server is a CoAP client for receiving CoAP messages, a REST server for communication with the operation management system, a

sensor and actuator node. It is composed of Mongo DB to store the resources provided by the user, and web server for user interface. The control node can be run on a PC using the Linux kernel because it does not need to be connected to the sensor and actuator modules, even though it can be also implemented in the Raspberry Pi.

The web server periodically reads data from Mongo DB. However, when real-time data is needed, it is obtained through the REST server. For convenience, the web server is implemented in the same hardware system (same node), but since the web server plays the role of the operation management system, it is logically mounted on a separate system.

The CoAP Client periodically requests data from the sensor and driver node (CoAP server), and stores the data obtained from the CoAP server in the DB. REST server gets real-time data through CoAP Client.

Figure 7 is a MongoDB collection for storing resource information generated by the sensor and actuator nodes of the smart farm testbed. The collection can be divided into a node on which sensor and actuator modules are mounted, and a collection for each sensor module. The node collection includes the ID as an identifier for classifying each device, the device type for classifying the sensor and actuator nodes, the node name as data, the address assigned to the node, the state of the node, and the storage time. The sensor collection includes ID as an identifier to classify devices, device type to classify devices, id of a node connected as data, sensor type, sensor value, sensor value unit, sensor status, sensor normality, storage time, etc.

**Figure 7.** Node and Sensor Collection

Some resource types were newly defined and utilized based on existing resource types for interworking with smart farm system devices.

1) CO₂ sensor: Define a new resource type based on air quality and CO₂ resource type. (The carbon dioxide concentration value is used in combination with the air quality resource type because the existing carbon dioxide sensor resource type can only display whether exceeds a certain concentration of CO₂.)

2) Relay: Define a new resource type based on the binary switch resource type (onewayrelay)

3) DC motor: Define a new resource type based on the existing binary switch resource type (dcmotor)

Figure 8 shows web server and REST server interworking. Interworking with the web server is to implement the operation management system, and to implement a pilot application that can be installed in the operation management

system. For example, in order to monitor the status of all nodes through the web server or check the values of nodes or sensors under a specific condition, REST API is organized as follows.

- /data/nodes: GET, request information of all nodes
- /data/nodes/sensor/type: GET, request information of selected sensor of all nodes
- /data/node/<int:id>: Request node information using GET, ID
- /data/node/<string:name>: GET, request node information by name
- /data/node/<int:id>/sensor/type: GET, request the sensor information of the target node (ID)
- /data/node/<string:name>/sensor/type: GET, request the sensor information of the target node (Name)
- /data/node/<int:id>/sensor/type/status: POST, sending device control information of the target node (ID) type (relay)
- /data/node/<string:name>/sensor/type/status: POST, sending device control information of the target node (Name) type (relay)

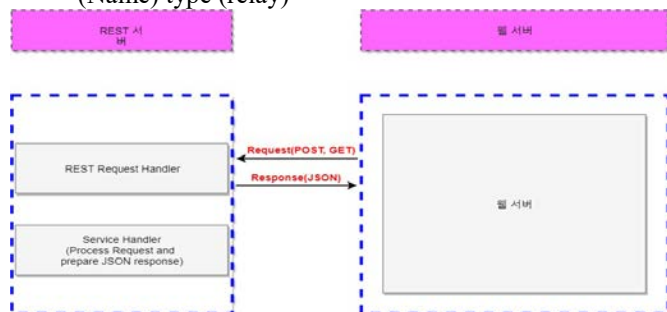


Figure 8. Web server and REST server interworking

IV. CONCLUSIONS

IoT devices used in smart farms frequently fail due to frequent water supply and high temperature and humidity, which causes errors in a significant amount of data. In order to solve this problem, many current smart farm device manufacturers are developing their own products with their own know-how.

However, if it is possible to manufacture open products that can guarantee mutual compatibility, each smart farm device manufacturer can focus solely on their own product development, which can lead to improved product competitiveness. In addition, it can be also expected to realize economies of scale through mass production.

In this paper, the proof-of-concept (PoC) for smart farm application was verified by implementing a smart farm testbed using the IoTivity, open source software provided by OCF. Through this, we checked the technical feasibility that general smart farm device manufacturers or smart farm operators can easily use and operate IOT devices.

The results of this paper are expected to be used for the spread of open smart farms that ensure compatibility between smart farm devices in the future.

ACKNOWLEDGMENT

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) through Advanced Production Technology Development Program Program(or Project), funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) (317015-6)

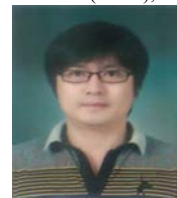
REFERENCES

- [1] ICT Standard & Certification Special Theme: Smart Farm, TTA Journal, vol.180, Dec. 2018.
- [2] Development of Korean Smart Farm Technology, Rural Development Administration, 2018.
- [3] Open Connectivity Foundation, <https://openconnectivity.org/>
- [4] IoTivity Project, <https://iotivity.org/>
- [5] S. H. Park, "OCF IoT Open-source Platform Standards Trend", TTA Journal Special Report, vol. 166, pp.41-45, July 2016.
- [6] W. S. Lee, H. Cha, J. H. Jeon, "IoT Open-source Technology – IoTivity", The Korean Institute of Communications and Information Sciences(KICS), Information and Communications Open Lecture, vol. 32, no 12, pp.27~35, Nov. 2015.
- [7] CoAP: An Application Protocol for Billions of Tiny Internet Nodes, Bormann, Carsten; Castellani, Angelo P; Shelby, Zach. IEEE Internet Computing 16.2, pp.62-67, Mar 2012

Sunghyun Yoon is a principal researcher of Protocol Engineering Center in Electronics and Telecommunications Research Institute (ETRI), Korea. He received a Ph.D. degree from the Chungnam National University, Korea in 2012. Since 2000, he joined ETRI and has performed many research projects in the area of network system engineering, network simulation, network evolution, network mobility, service management, unified AAA architecture, trusted reality, smart IoT platform and so on. Currently, he is involved mainly in the development of standards for interoperable smart farm ICT devices. His research interests include edge cloud based smart farm management and control platform, web based smart farm data model and smart farm service API.



Mun-Hwan Choi is a senior researcher of Protocol Engineering Center in Electronics and Telecommunications Research Institute (ETRI), Korea. He received a Ph.D. degree from the Kunsan National University, Korea in 2008. Since 2007, he joined ETRI and has performed many research projects in the area of local communication network, terminal device conformity assessment and smart farm so on. His research interests include telecommunication technical criteria, terminal device technology standards and smart farm device interoperability.



Juyoung Park is a principal researcher of Protocol Engineering Center in Electronics and Telecommunications Research Institute (ETRI), Korea. He received a Ph.D. degree from the Chungnam National University, Korea in 2001. Since 2001, he joined ETRI and has performed many research projects in the area of Multicast, QoS protocol and architecture, IoT, Smart Agriculture, Wireless power transmission. He is also an international standards expert of ITU-T SG11, SG13, SG20 and IEC TC100.

