

# Implementing Autonomous Environmental Sensing in Smart City with IoT-Based Sensor Box and Cloud Services

Seiji Sakakibara<sup>\*</sup>, Sachio Saiki<sup>\*</sup>, Masahide Nakamura<sup>\*</sup>,  
Shinsuke Matsumoto<sup>†</sup>

## Abstract

To realize indoor environmental sensing, which is a key technology of providing smart services in smart city, with low cost, our research group has proposed a small IoT device named *sensor box*. In the previous sensor box, however, it is difficult to deploy for the smart city with some problems. In this paper, we propose an indoor environment sensing service using autonomous sensor box to adapt the previous sensor box for the smart city. To confirm the effectiveness of proposed service, we deploy autonomous sensor boxes on practical indoor environments.

*Keywords:* Smart City, IoT, Environmental Sensing, Sensor Box

## 1 Introduction

*Smart city*[1][2] is a next-generation social system, which provides residents value-added services by collecting and managing various data with ICT or environmental engineering. Currently, as one of the approaches for realizing the smart city, energy optimization service is putting into practical use with HEMS[6]. Moreover, thanks to the progress of IoT[7] in recent years, many indoor devices can be connected to the Internet, so a smart home, which improves life quality, is attracting much attention. Going forward, health care, aged care, crime prevention and disaster prevention services are expected. To provide these smart services, it is important to collect various indoor data not limited to energy and grasp the indoor environment.

*Indoor environment sensing* is a method to measure, accumulate and utilize indoor environment data with various sensing devices. Indoor environment sensing is a key technology for providing smart services, but a cost of infrastructure such as deployment of environment sensors and preparations of database has become a problem. To realize indoor environment sensing with low cost, our research group has proposed a small IoT device named *sensor box*[3]. With deploying this sensor box for each house in the smart city and integrating data to Cloud, indoor environment sensing can be realized without cost-intensive infrastructure.

---

<sup>\*</sup> Kobe University, Hyogo, Japan

<sup>†</sup> Osaka University, Osaka, Japan

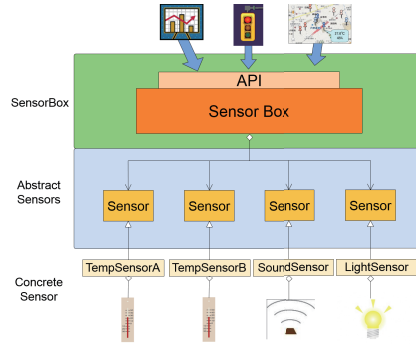


Figure 1: Architecture of SensorBox Framework

However, when we apply this sensor box to the smart city directly, next P1-P3 problems occur.

- P1: Configuring of a sensor box relies on manual works.
- P2: Measured data does not link with measurement environment information.
- P3: Collected data cannot be utilized for an application.

With P1, we take high cost when we deploy the sensor box on a scale of a smart city scale. With P2, we cannot judge where the measurement data is obtained. With P3, we cannot utilize large-scale sensor data as value-added services.

In this study, to solve these three problems, we propose new indoor environment sensing service in the smart city. In particular, to solve P1 first, we develop an *autonomous sensor box* which operates autonomously. With electric power and network, autonomous sensor box configures itself automatically in cooperation with management service on Cloud. Then autonomous sensor box uploads measured environmental data to Cloud. Next, to solve P2, we cooperate with **Scallop4SC**[4], which our research group has proposed. With Scallop4SC, we link environmental data measured by the Sensor Box with information data such as location or owner. Last, to solve P3, we define new *standard schema* of data which the Sensor Box sends to Cloud. In addition, we develop *Web services* for various services to get sensor log or sensor status via Web-API.

In this paper, we made an actual autonomous sensor box with Raspberry Pi and Phidgets sensors. Also, we deployed the autonomous sensor boxes in practical indoor environments and executed indoor environment sensing. As a result, we confirmed that we can execute indoor environment sensing without cost-intensive infrastructure and configuration labor.

## 2 Preliminaries

### 2.1 Sensor Box Framework

Before now, sensor applications have been studied and developed in various fields. However, in most existing applications, software and sensor devices are tightly coupled. Therefore, replacement of devices is impossible, and sensor reusability and data transferability are very poor. To solve these problems, our research group has proposed *sensor box framework*[3]. Sensor box framework is an application framework, which abstracts sensor devices, and manages sensors as a unit of box by combining multiple types of sensors freely. Therefore, various applications can share the sensors. Figure 1 shows its architecture. In

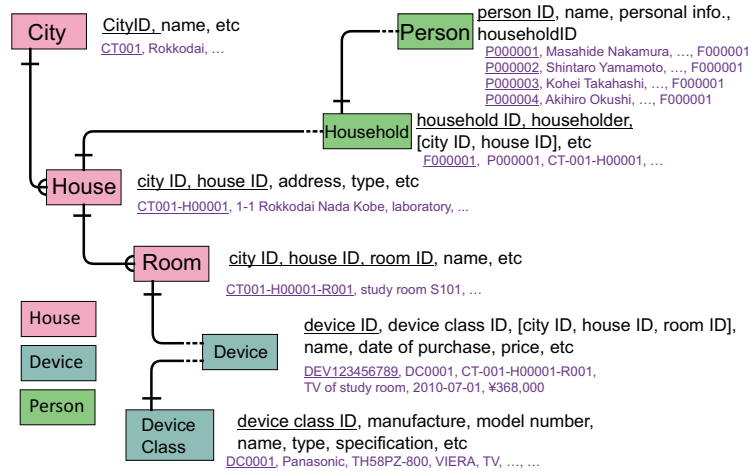


Figure 2: Data Model of Smart City Configuration

this picture, the top level layer exposes sensor box API as a Web service. Then various applications can utilize sensor box in a cross-sectional way, using standard Web technology. In this framework, various sensor devices can be all used as Web services, so sensors take the place of IoT virtually.

In sensor box framework, to give specific sensor information to the framework, *sensor box definition file* is needed. The sensor box owner makes definition file manually and gives to the framework. The framework loads definition file at runtime and creates proper sensor object using reflection. Replacement of sensor devices can be realized by rewriting of the definition file.

## 2.2 Scallop4SC

Scallop4SC (SCALable LOGgingPlatform for Smart City)[4] is a data processing platform, which accumulates and utilizes various big data log obtained in the smart city. Scallop4SC has following four key functions, F1–F4. **F1**: Collecting and storing various logs of the smart city. **F2**: Efficient processing of stored logs. **F3**: Unified management of smart city configuration information. **F4**: Providing general access APIs to the above data.

In the third function, Scallop4SC manages *configuration information* about what devices are deployed and where the devices are deployed in the smart city. Figure 2 shows the Scallop4SC configuration information data schema, which we designed in the previous study[4]. This Entity Relationship Diagram based on notation shown in the reference[5]. In the figure, smart city configuration information of smart city comprises three information: *House information*, *Device information*, *Person information*.

## 2.3 Indoor Environment Sensing for Smart City

An indoor environment sensing is a method to measure various indoor environmental data using environment sensors. Environment data obtained by an indoor environment sensing have been utilized in the closed environment. However, various smart city services which optimize city function are expected with *smart city indoor environment sensing* gathering and managing city-scale environment data.

Indoor environment sensing of each house in the city can be realized relatively easily with sensor box and framework as stated 2.1. However, the previous sensor box framework does not be assumed data gathering in smart city scale. Therefore, application to smart city indoor environment sensing has following challenges. (Each challenge corresponds to P1-P3 as stated 1.) For P1, configuring of the sensor box relies on manual works, for P2, measured data does not link with measurement environment information and for P3, collected data cannot be expressed and utilized.

The first challenge, human cost for sensor box configuration become huge. The second challenge, location of measured environment data is unknown. The third challenge, collected data cannot be utilized as valuable services for the citizen. Then, in this study, we propose new *smart city indoor environment sensing service*, which meets next three demands.

**R1:** Minimize the configuration and deployment and management cost.

**R2:** Link measured data to smart city configuration information.

**R3:** Define data accumulate base and data schema.

To meet these three demands, in this study, we develop new Sensor Box (*autonomous sensor box*), which executes environment sensing autonomously and develop supporting Cloud Service.

### 3 Proposed Method

#### 3.1 Entire Architecture

Figure 3 shows the entire architecture of smart city indoor environment sensing service in this study. The proposed service roughly comprises follow S1-S4 elements.

5 Cloud service, which keeps measured new Sensor Box data to memory.

**S1: Autonomous sensor box**

With power and network connection, execute indoor environment sensing autonomously and upload data to Cloud.

**S2: Sensor Box Management Service**

Manage all Sensor Box definition file in smart city and configuration information.

**S3: Log Collection Service**

Collect centrally measured Sensor Box data to large-scale database as time series data.

**S4: State Cache Service**

Cloud service, which keeps measured new Sensor Box data to memory.

In the proposed service, cooperate and integrate above four elements by *Service-Oriented Architecture (SOA)*, then realize smart city indoor environment sensing service. Each element is described in the following sections.

#### 3.2 Autonomous Sensor Box

To execute indoor environment sensing, we use Sensor Box. The hardware of Sensor Box comprises a number of *environment sensor (device)* and *sensor hub*, which control sensors. As the S1 of the figure 3 shows, sensor box framework as stated 2.1 is installed into a sensor hub. Logger application in the sensor hub, calls Sensor Box Web service(REST) periodically and get sensor data, then upload to Cloud as JSON format text. To use sensor

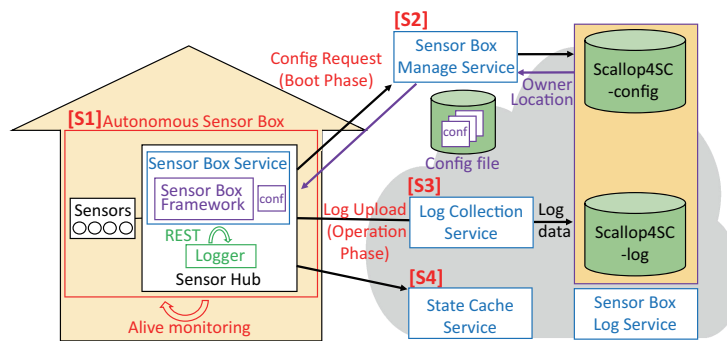


Figure 3: System Architecture of Proposed Service

box framework, need sensor box configuration file (section 2.1), which describe connected environment sensors. In the previous sensor box, the owner had to set definition file to the sensor box manually, when the number of the box increase, configuration and management cost is large. Then, in the autonomous sensor box, manage definition file on Cloud centrally. When Sensor Box boot, Sensor Box download appropriate definition file (reference conf and S2 in the figure3). With central management of definition file on Cloud, when change sensor device type or installation location, only need change information of definition file on Cloud. Also, to identify each Sensor Box in the smart city, give each Sensor Box unique ID (*sensor box ID*). Moreover, for each Sensor Box, register previously sensor box ID, installation location, and owner information to the Scallop4SC (reference 2.2). Thus, link measured data described below, with measure environment information.

The key functions of autonomous sensor box are as follows.

#### **Auto connect to the network:**

When Sensor Box powers up, connect to the network automatically and prepare for connection to Cloud.

#### **Auto configuration of Sensor Box:**

When to connecting to the Cloud, configure Sensor Box automatically. In the beginning, Sensor Box gives ID of itself to Sensor Box Manage Service. Sensor Box Manage Service, based on given ID, acquires the smart city configuration information from the Scallop4SC. Then, Sensor Box Manage Service add-on configuration information to the appropriate definition file and return to the Sensor Box.

#### **Auto launch logger:**

After Sensor Box configuration, launch sensor logger automatically. The logger uses Web service API of sensor box framework and gets connected sensors value. Logger uploads acquired values to two type Cloud service (Log Collection Service, State Cache Service). Acquisition and upload of sensor value are taken place by given interval.

#### **Alive monitoring of logger:**

While Sensor Box runs, make sure logger is alive for the constant interval, and if the logger is stopped, reboot automatically.

With above autonomous function, the user can begin environment sensing automatically only by deploying Sensor Box and giving power. Then minimize the configuration and

**Table 1: Data Schema of Environmental Sensor Data**

	Key name	Value description	Key-Value example
data	light	Measured sensor values	light:82
	temperature		temperature:10.667
	...		...
info	date	The date of log obtained	date:2016-02-04
	timeOfDay	The time (of day) of log obtained	timeOfDay:16:07:39
	time	The time of log obtained	time:2016-02-04-T16:07:39+09:00
	boxid	Sensor Box ID	boxid:sbox-phidget-406364
	owner	Owner of Sensor Box	owner:sakakibara
	location	Installation location of Sensor Box	location:Kobe/Kobe-Univ./S101/desk

management cost at deployment.

### 3.3 Sensor Box Manage Service

Sensor Box Manage Service is Cloud service, which cooperate with Scallop4SC and manage all Sensor Box in the smart city. As the S2 of the figure 3 shows, manage service is accessed when each Sensor Box boot, acquires smart city configuration information from Scallop4SC based on given ID. Then, add-on smart city configuration information to the definition file of the appropriate Sensor Box, and return the file to the Sensor Box.

Furthermore, Sensor Box Manage Service manages network connection information (IP address and others) of each Sensor Box and use this information to remote test or maintenance.

### 3.4 Log Collection Service

Log Collection Service is a service, which collects environment data measured by Sensor Box and accumulates as time series data (S3 of figure 3). To realize efficient search and aggregation of sensor data, Log Collection Service suppose data schema as shown in the table1. In the table, “data” shows sensor value measured by Sensor Box, “info” shows meta information which describes sensor value.

To deal centrally measured data from Sensor Box which have various sensor combination, does not define the strict schema of sensor data itself, express Key-Value of attribute name and value.

On the other hand, as to metadata, define the common attribute which does not depend on sensor type, and utilize for search and aggregation of sensor data. Specifically, clarify items of metadata from 3 viewpoints: when, who, where. These viewpoint does not depend on specific environment sensing. “data” and “timeOfDay” and “time” describes the time when the logger obtain sensor value. “boxId” and “owner” describe who obtain sensor value, deal sensor box ID and owner information acquired from Scallop4SC. “location” describe where Sensor Box obtains the value, deal Sensor Box installation information acquired from Scallop4SC.

Logger in Sensor Box, in every time of measurement, create data based on the schema in the table1 and modify to JSON format text, then upload to the Log Collection Service. Collected time series sensor data is used by various applications through Sensor Box Log Service (bottom right corner of the figure 3).

### 3.5 State Cache Service

State Cache Service is a service (S4 in the figure3), which saves only the latest data sent one after another from Sensor Box. Then applications can access the current value of Sen-

sensor Box quickly. Time series data accumulated in the Log Collection Service is useful for the application which uses past values. However, for the application only need the current value, overhead to search latest data from accumulated data increase. State Cache Service, sensor box ID as “Key” and current value as “Value”, save latest measured value on memory, then realize quick access to the current value of any Sensor Box. With upload measured sensor value to both Log Collection Service and State Cache Service, autonomous sensor box realizes efficient data provision to both the application which utilizes the past and current data.

### 3.6 The Entire Flow of the Service

In indoor environment sensing service for the smart city, before deploying Sensor Box, make sensor box definition file which describes used device information on the Sensor Box Manage Service, and registers installation location or owner information as smart city configuration information to the Scallop4SC. When deployed Sensor Box boot, Sensor Box gives own ID to Sensor Box Manage Service on Cloud. Sensor Box Manage Service call Scallop4SC by given ID, create sensor box definition file based on smart city configuration information, and return to Sensor Box Service. Sensor box framework configures sensor device based on acquired sensor box definition file. Logger receives measured value through Sensor Box Service and forms data according to the schema. After that, upload data to the Log Collect Service and State Cache Service on Cloud as JSON format text data. Log Collection Service accumulate measured data to Cloud database, and State Cache Service save only current data.

### 3.7 Sufficiency Requirements

We make sure that the proposed service meets requirements stated 2.3.

- **R1: Minimize the configuration and deployment and management cost.**  
Indoor environment sensing can be executed only by deploying Sensor Box to the house. Sensor Box acquires definition file for itself from Cloud, when change type of sensor device and installation location, it can be dealt with only updates the management information on Cloud.
- **R2: Link measured data to smart city configuration information.**  
By registering Sensor Box installation information to the Scallop4SC, we can create sensor box definition file added smart city configuration information. Thus, we can link measured data to smart city configuration information.
- **R3: Define data accumulate base and data schema.** By the proposed data schema, define collected sensor data structure. Also, accumulated time series data and current value are utilized by various applications through Log Collection Service and State Cache Service.

## 4 Prototype Implementation

### 4.1 Prototyping

We implemented a prototyping and conducted an experiment. The purpose of the experiment is to confirm sufficiency of three requirements in 3.7 in actual environment.

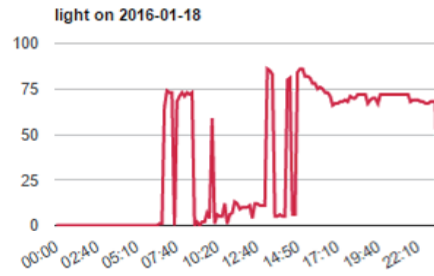
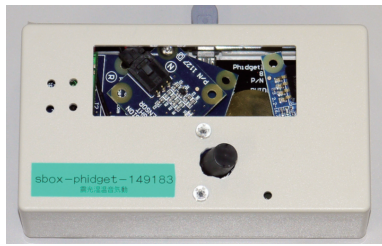


Figure 4: Prototype of SensorBox

Figure 5: Screenshot of SensorBoxLogService

**Autonomous sensor box (S1):** Figure 4 shows the appearance of the autonomous sensor box. As housing sensor devices, we used seven types of environment sensor produced by Phidgets[8]: Temperature Sensor (Humidity/Temperature Sensor 1125), Humidity Sensor (Humidity/Temperature Sensor 1125), Atmosphere Pressure Sensor (Absolute Pressure Sensor 1141), Vibration Sensor (Vibration Sensor 1104), Sound Sensor (Sound Sensor 1133), Light Sensor (Light Sensor 1127), Motion Sensor (Motion Sensor 1111).

These seven types of sensors connect with the sensor hub through PhidgetInterfaceKit, which is an interface board. For sensor hub, we used Raspberry Pi 2 (Model B, Raspbian Jessie), which is a one-board microcomputer. In the figure 4, the case contains environment sensors and interface board, and USB cable is connected to the Raspberry Pi. We implemented sensor logger in Perl and alive monitoring system in shell and cron. Also, we implemented Sensor Box Service in Java and deploy on the Raspberry Pi as Web service using Apache Axis2.

**Sensor Box Manage Service (S2):** We implemented Sensor Box Manage Service API in Perl CGI and HTML::Template library. We developed API connection of Scallop4SC in Java, and database in MySQL, and O/R mapper in MyBatis.

**Log Collection Service (S3):** We implemented Log Collection Service in Fluentd, which is log collection framework. For the database, we used MongoDB and HBase.

**State Cache Service (S4):** We developed in Java Map and deployed as RESTful Web service using Jersey.

## 4.2 Deployment in Actual Environment

To confirm the practicality of the proposed service, we deployed Sensor Box on the house of study collaborators and doing long-term environment sensing. We deployed three Sensor Boxes, doing environment sensing in the living room, entrance, and washroom. Figure 5 shows the screenshot of an application, which visualizes acquired data by Log Collection Service. The figure shows part of measured data in the living room on January 18, 2016, as time series data. For example, in the light of the items, the value measured by Light Sensor is displayed as time series in lux. We show an output example of State Cache Service Log as following. The current measured value of Sensor Box “sbox-phidget-406364” is displayed in the info field and data field. This data schema is based on table1 in 3.4. This shows that latest measured data is linked to smart city configuration information as JSON format. We know that owner is “yasuda.kiyoshi”, deployed location is “CT002/H00001/R003/shoebox” and latest temperature is about “10.2”. Even though collaborator of this study does not have



ICT knowledge, Sensor Box can be operated without requesting the work of setting and maintenance.

StateCacheService Log

```
"sbox-phidget-406364":{"info":{"boxid":"sbox-phidget-406364","date":"2016-02-10","location":
"CT002/H00001/R003/shoebox","owner":"yasuda.kiyoshi","time":"2016-02-
10T21:06:42+09:00","timeOfDay":"21:06:42", "data":{"light":"0","vibration":"500.0",
"temperature":"10.222620000000006","motion":"false","gasPressure":"99.7824347826087",
"sound":"0.0","humidity":"49.38199999999999","presence":"0"}}
```

### 4.3 Evaluation

From the aspect of R1, we can get normal sensing results in actual usage through non-specialist user's environment. On the basis of this result, we can conclude that easy installation and operation can be realized by use of autonomous sensor box. In several times, we notice that the Sensor Box cannot send sensing data correctly because of the software error of Raspberry Pi or network. In this case, we could continue sensing by requesting for the user to turn off the power and then turn it again. Even if the Sensor Box gets critical error such as hardware broken, a new Sensor Box can be adapted easily to the existing environment. Additionally, we need an abnormality detection method of Sensor Box to give the lower maintenance cost. For R2, now we are operating 20 Sensor Box, and all of them can accumulate data, which linked to smart city configuration information. For R3, all data is accumulated based on proposed data schema, and we can utilize service such as Sensor Box Log Service.

## 5 Related Works

To realize environment sensing on a small scale, many companies have developed sensor device with various environmental sensors[9][10]. Unfortunately, these devices cannot adapt to large-scale sensing required by smart city because of the lack of functions. In other words, we have to develop many additional functions to these devices for realizing the large-scale sensing. Therefore, it is difficult to realize the smart city in an easy way compared with our proposed Sensor Box. On the other hand, for easy environment sensing, some managed sensing services have been provided by some companies[11][12]. These services give sensing data, data viewer, alerts and notifications based on sensing results for users. However, these service targets for only specified user, gathering sensing results of multi-home or building are not considered. For the reasons stated above, our proposed method is more suitable than other solutions for the smart city realization.

## 6 Conclusion

In this paper, we proposed new indoor environment sensing service for smart city. In the proposed service, Sensor Box contained environment sensors execute autonomously environment sensing without human work, and collect data to Cloud, cooperating with three Cloud services. Also, we developed the prototype of the proposed service and conducted the test with deploying in the actual house. Thus, we confirm that environment sensing can be executed continuously without requesting the work of setting or maintenance. As the

future challenges, we have to develop value-added service utilizing collected indoor environment data in the smart city. Moreover, not only stationary type of Sensor Box proposed in this study, cooperation with mobile type sensors such as a smartphone or wearable device is the interesting theme.

## Acknowledgments

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (No.26280115, No.15H02701), Young Scientists (B) (No.26730155), and Challenging Exploratory Research (15K12020) ].

## References

- [1] Paolo Neirotti and Alberto De Marco and Anna Corinna Cagliano and Giulio Mangano and Francesco Scorrano, “ Current trends in Smart City initiatives: Some stylised facts,” *Cities*, vol. 38, 2014, pp. 25-36.
- [2] Robert G. Hollands, “ Will the real smart city please stand up?,” *City*, vol. 12, no. 3, 2008, pp. 303-320.
- [3] Akihiro OKUSHI, Shinsuke MATSUMOTO, Sachio SAIKI, Masahide NAKAMURA, “ A Framework for Personal Sensor Box in Mobile Environmental Sensing,” *IEICE Technical Report*, vol. 113, no. 327, 2013, pp. 51-56.
- [4] Shintaro YAMAMOTO, Shinsuke MATSUMOTO, Masahide NAKAMURA, “ Implementing API of Large-scale House Log Data Platform in Smart City,” *IEICE Technical Report*, vol. 112, no. 305, 2012, pp. 27-32.
- [5] Kozo Watanabe, “ Hanbai Kanri System de Manabu Modeling Koza,” *Shoeisha*, 2008(in Japanese).
- [6] Young-Sung Son and Pulkkinen, T. and Kyeong Deok Moon and Chaekyu Kim, “ Home energy management system based on power line communication,” *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, 2010, pp. 1380-1386.
- [7] Jayavardhana Gubbi and Rajkumar Buyya and Slaven Marusic and Marimuthu Palaniswami, “ Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645-1660.
- [8] Phidgets Inc. “ Unique and Easy to Use USB Interfaces,” <http://www.phidgets.com/>.
- [9] Texas Instruments Inc. “ SensorTag,” [http://www.tij.co.jp/ww/wireless\\_connectivity/sensortag/](http://www.tij.co.jp/ww/wireless_connectivity/sensortag/).
- [10] MATRIX Labs. “MATRIX Creator on a Raspberry Pi,” <https://creator.matrix.one/>.
- [11] OMRON Corporation. “ Environment Sensor,” <https://www.components.omron.com/solutions/mems-sensors/environment-sensor>.
- [12] MouseComputer CO., LTD. “ mouse smart home,” <http://www.mouse-jp.co.jp/smarthome/>.