|  |
| --- |
|  |

|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | RDM#53 |
| Title:\* | Use case on IoT device calibration |
| Source:\* | JaeSeung Song, KETI, jssong@sejong.ac.kr  Minbyeong Lee, Hyundai Motors, [minbyeong.lee@hyundai.com](mailto:minbyeong.lee@hyundai.com)  Franck Le Gall, EGM, [Franck.le-gall@egm.io](mailto:Franck.le-gall@egm.io) |
| Date:\* | 2022-02-11 |
| Input related to\* | TR-0001 (Rel-5) |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Impacted other TS/TR(s) |  |
| Decision requested or recommendation:\* | Agree for inclusion in TR-0001 |
| Template Version: January 2017 (Do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

# Introduction

This contribution introduces a new use case for supporting IoT device calibration via IoT service layer platform.

* Update section number (target to the Enterprise Use Cases)
* A group calibration use case will be prepared as a separate use case 🡪 No action required in TP #53
* Add a paragraph describing an example for CO2 sensors

### -----------------------Start of change 1-------------------------------------------

## 6.x. Use case #x – IoT device calibration

### 6.x.1 Description

IoT sensors measure physical value and convert the measurement to a digital value. For example, a temperature sensor measures the temperature of a place where it locates. Ideally, the same type of sensors manufactured by the same manufacturer should measure the same value if deployed at the same place. However, depending on the characteristics of the sensor device, the measured value may be different. Also, there is a case that a sensor does not have a proper zero reference value, which generates a wrong measurement value.

The sensor’s range may shift due to the same conditions just noted, or perhaps the operating range of the process has changed. For example, a process may currently operate in the range of 0 to 200 Celsus, but changes in operation will require it to run in the range of 0 to 500 Celsus.

In the case of CO2 sensors, it is essential to know in which domain the sensors measure the percentage of carbon dioxide. Based on such information, the range, accuracy, or precision can be decided to recommend the proper carbon dioxide. For example, indoor and outdoor air has between 400ppm and 2,000ppm CO2 by volume. This means that for measuring a smart home's indoor air quality, a CO2 sensor requires different calibration equations to apply.

Therefore, each sensor needs to be calibrated before deployment or adjustments for the measured value should be applied.

In order to support calibration, the sensors typically have a separate circuit and memory space internally. For example, the sensors should support a logic to perform calibration and space to store equations for calibration. These resources increase the price of sensors and make it challenging to build a small and cheap sensor.

The basic concept of this use case is to allow IoT platforms to support the calibration of their managing sensors. Typically, when a sensor is deployed for the first time, it needs calibration with a local test machine, and the machine generates results for calibration. The generated calibration value is stored in the cloud IoT platform, where the sensor will be registered and managed. In this case, the sensor does not need to provide any computing power or memory for calibration. When a sensor registers to the IoT platform, the platform checks a corresponding calibration value for the sensor and store such value as an additional parameter to the resource for the sensor. When the sensor generates a measurement, the platform checks the calibration value, and if there exist, the measurement will be adjusted based on the calibration value.

### 6.x.2 Source

### None

### 6.x.3 Actors

* Calibration application: a testing device to calibrate IoT devices.
* IoT platform: An IoT platform stores data for device calibration and performs calibration when there is a new measurement.

### 6.x.4 Pre-conditions

* The sensor device does not have enough computing power and memory to store and perform calibration.
* The IoT platform holds a set of information to support device calibration.

### 6.x.5 Triggers

* If the IoT platform is configured to support IoT device calibration, a new measurement from an IoT sensor that needs calibration triggers the process for IoT device calibration.

### 6.x.6 Normal Flow

Figure 6.x.6-1 illustrates the high-level flows of the IoT sensor calibration use case, which consists of the following steps:

* Step 1: The calibration application retrieve calibration parameters for the type of Sensor-1 from the IoT platform.
* Step 2: The calibration application calibrates Sensor-1 using the retrieved parameters and generates calibration results.
* Step 3: The calibration application stores the generated calibration results for Sensor-1.
* Step 4: Sensor-1 registers itself to the IoT platform.
* Step 5: The IoT platform checks the calibration data for Sensor-1 and stores it as parts of associated parameters.
* Step 6: Sensor-1 sends a request to store its new measurement to the IoT platform.
* Step 7: The IoT platform adjusts the measurement based on the calibration results and stores adjusted value.



Figure 6.x.6-1 A flow for calibrating IoT devices in the server IoT platform

### 6.x.7 Alternative Flow

None

### 6.x.8 Post-conditions

The IoT platform stores the correct measurement after adjusting calibration value.

### 6.x.9 High Level Illustration



Figure 6.x.9-1 Conceptual diagram of IoT device calibration

### 6.x.10 Potential Requirements

1. The oneM2M System shall be able to store calibration parameters for an IoT device and calibrates new measurement.

### -----------------------End of change 1-------------------------------------------