|  |  |
| --- | --- |
| Project | **Specification of Sensor Interface for Cyber and Physical World**<<https://sagroups.ieee.org/2888.1/> **>** |
| Title | **Application programming interfaces for environment-related smart sensors** |
| DCN | **2888-21-0071-00-0001** |
| Date Submitted | **Oct. 13th, 2021** |
| Source(s) | Sang-Kyun Kim, goldmunt@gmail.com (Myongji University)Min Hyuk Jeong, jmh8900@gmail.com (Myongji University) |
| Re: |  |
| Abstract | This contribution illustrates the application programming interfaces for environment-related sensors. |
| Purpose | To start discussion on purpose of the standard |
| Notice | This document has been prepared to assist the IEEE 2888 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 2888 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

# Introduction

This contribution illustrates the application programming interfaces for environment-related sensors.

## Ambient light sensor

Table 1 – Ambient light sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| AmbientLight() |
| *Default constructor.* |
|  |
| AmbientLight(String id) |
|  |
| AmbientLight(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getAmbientLightSensorData() |
|  | *This function returns sensor data from an ambient light sensor in JSON format.* |
|  |  |

## Ambient noise sensor

Table 2 – Ambient noise sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| AmbientNoise() |
| *Default constructor.* |
|  |
| AmbientNoise(String id) |
|  |
| AmbientNoise(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getAmbientNoiseSensorData() |
|  | *This function returns sensor data from an ambient noise sensor in JSON format.* |
|  |  |

## Temperature sensor

Table 3 – Temperature sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Temperature() |
| *Default constructor.* |
|  |
| Temperature(String id) |
|  |
| Temperature(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getTemperatureSensorData() |
|  | *This function returns sensor data from a temperature sensor in JSON format.* |
|  |  |

## Humidity sensor

Table 4 – Humidity sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Humidity() |
| *Default constructor.* |
|  |
| Humidity(String id) |
|  |
| Humidity(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getHumiditySensorData() |
|  | *This function returns sensor data from a humidity sensor in JSON format.* |
|  |  |

## Wind sensor

Table 5 – Wind sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Wind() |
| *Default constructor.* |
|  |
| Wind(String id) |
|  |
| Wind(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getWindSensorData() |
|  | *This function returns sensor data from a wind sensor in JSON format.* |
|  |  |

## Gas sensor

Table 6 – Gas sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Gas() |
| *Default constructor.* |
|  |
| Gas(String id) |
|  |
| Gas(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getGasSensorData() |
|  | *This function returns sensor data from a gas sensor in JSON format.* |
|  |  |

## Dust sensor

Table 7 – Dust sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Dust() |
| *Default constructor.* |
|  |
| Dust(String id) |
|  |
| Dust(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getDustSensorData() |
|  | *This function returns sensor data from a dust sensor in JSON format.* |
|  |  |

## Weather

Table 7 – Weather sensor API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Weather() |
| *Default constructor.* |
|  |
| Weather(String id) |
|  |
| Weather(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| JSONObject | getWeatherSensorData() |
|  | *This function returns sensor data from a weather sensor in JSON format.* |
|  |  |