|  |  |
| --- | --- |
| Project | **Standard for Actuator Interface for Cyber and Physical World**<https://sagroups.ieee.org/2888/ **>** |
| Title | **Revised Root Class Example and Data Format for Individual Actuator Command** |
| DCN | **2888-22-0041-00-0002** |
| Date Submitted | **Feb. 14, 2022** |
| Source(s) | Yegi Lee zxcasd312@naver.com (Konkuk University)Shin Kim new.xin22@gmail.com (Konkuk University)Kyoungro Yoon yoonk@konkuk.ac.kr (Konkuk University) |
| Re: |  |
| Abstract | This contribution proposes the revised root class and data format for individual actuator command.  |
| 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 proposes the revised root class example and data format for individual actuator commands. During the 9th meeting, we discussed the structure of the root class and decided that it is necessary to revise it. The general, syntax and semantics of the root class are not changed, but the example is changed in this contribution. So, we propose the updated version of the data format for individual actuator commands according to our consensus.

# Data formats for interfacing actuator command

* 1. Root schema structure
		1. General

This clause defines the root schema structure for interfacing actuator command. This root schema structure specifies tools for describing actions that each individual actuator is supposed to take.

* + 1. Syntax

|  |
| --- |
| { "$schema": "http://json-schema.org/draft-07/schema#", "title": "command data", "description": "Schema for command data", "type": "object", "properties": { "timeStamp": { "type": "string", "format": "date-time" }, "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, // This is where the properties of each type of co data are declared by reference. // For Example,// "lightCommandData": {// "$ref": "#/definitions/lightCommandData"// },// "flashCommandData": {// "$ref": "#/definitions/flashCommandData"// },// …},"additionalProperties": false,"minProperties": 1,"maxProperties": 2,"definitions": {"commandInfoBaseAttributes": { "additionalProperties": false, "type": "object", "properties": { "id": { "type": "string" }, "actuatorIdRef": { "type": "string" }, "activate": { "type": "boolean" } } },// This is where the properties of each type of command data are defined. // For Example,// "lightCommandData": {// …// },// "flashCommandData": {// …// },// …}} |

* + 1. Semantics

The semantics of the root element.

| *Name* | *Definition* |
| --- | --- |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| timeStamp | Provides information about the time the actuator data command. |
| id | Uniquely identifies individual command data. |
| actuatorIdRef | Provides a link to the actuator that generated the data contained in this particular instance. |
| activate | Provide to command to turn on and off the device or actuator. A value of "true" means the actuator is commanded to activate, and "false" means the actuator is commanded to deactivate. In binary representation, a value of "1" means the actuator is commanded to activate, and "0" means the actuator is commanded to deactivate. |

* + 1. Examples

In this example, a light actuator commands the intensity of 100 lux with the color “blue” with the timestamp that means the time when the actuator data command is made. This light actuator is active. The command ID of this example is “C00000001” and the actuatorIDRef is “DC-01”.

|  |
| --- |
| { "timeStamp": "2018-11-13T20:20:39+00:00","lightCommandData": {"commandInfoBaseAttributes": { "id": "C00000001", "actuatorIdRef": "DC-01", "activate": true }, "color": "blue", "intensity": 100 }} |

* 1. Data Formats for Individual Actuators
		1. Light actuator
			1. General

This sub-clause specifies the actuator command type which can generate a light effect.

* + - 1. Syntax

|  |
| --- |
| "lightCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "color": { "$ref": "#/definitions/colorType" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false }, |

* + - 1. Semantics

The semantics of the lightCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| lightCommandData | Provide a structure for describing a command for a light actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| color | Describes the color that the light actuator can provide either as a reference to a term that shall be using the colorType. |
| intensity | Describes the intensity value that a light actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Examples

In this example, a light actuator commands the intensity of 120 lux with the color “blue”

|  |
| --- |
| { "lightCommandData": { "commandInfoBaseAttributes": {}, "color": "blue", "intensity": { "unit": "lux", "value": 120 } }} |

* + 1. Flash actuator
			1. General

This sub-clause specifies an actuator command type which can generate a flash effect.

* + - 1. Syntax

|  |
| --- |
| "flashCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "color": { "$ref": "#/definitions/colorType" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" }, "frequency": { "$ref": "#/definitions/genericCommandType/genericFrequencyType" } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the flashCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| flashCommandData | Provide a structure for describing a command for a flash actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| color | Describes the color that the flash actuator can provide either as a reference to a term that shall be using the colorType. |
| intensity | Describes the intensity value that a flash actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |
| frequency | Describes the frequency value that a flash actuator can generate. The frequency value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericFrequencyType. If the frequency unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Example

This example shows the description of the actuator command of flash effect with the following semantics. This flash actuator is commanded to perform the light intensity is 80 lux and frequency is 120 hz.

|  |
| --- |
| { "flashCommandData": { "commandInfoBaseAttributes": {}, "color": "blue", "intensity": { "unit": "lux", "value": 80 }, "frequency": { "unit": "hz", "value": 120 } }} |

* + 1. Scent actuator
			1. General

This sub-clause specifies an actuator command type which can generate a scent effect.

* + - 1. Syntax

|  |
| --- |
| "scentCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "scent": { "$ref": "#/definitions/scentType" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false, "required": [ "scent" ] }, |

* + - 1. Semantics

Semantics of the scentCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| scentCommandData | Provide a structure for describing a command for a scent actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| scent | Describes the scent that specific scent actuator can generate. The scent unit of the command value describes as a reference to a term that shall be using the scentType. |
| intensity | Describes the intensity value that a scent actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Examples

This example shows the description of an actuator command of scent effect with the following semantics. This scent actuator is commanded to perform the intensity 5% of the maximum intensity with the scent “rose”.

|  |
| --- |
| { "scentCommandData": { "commandInfoBaseAttributes": {}, "scent": "rose", "intensity": { "unit": "percentage", "value": 5 } }} |

* + 1. Heating actuator
			1. General

This sub-clause specifies the actuator command type which can generate a heating effect.

* + - 1. Syntax

|  |
| --- |
| "heatingCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the heatingCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| heatingCommandData | Provide a structure for describing a command for a heating actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| intensity | Describes the intensity value that a heating actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Examples

This example shows the description of an actuator command of heating effect with the following semantics. This heating actuator is commanded to perform the intensity of 42 celsius of the maximum intensity.

|  |
| --- |
| { "heatingCommandData": { "commandInfoBaseAttributes": {}, "intensity": { "unit": "celsius", "value": 42 } }} |

* + 1. Cooling actuator
			1. General

This sub-clause specifies the actuator command type which can generate a cooling effect.

* + - 1. Syntax

|  |
| --- |
| "coolingCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the coolingCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| coolingCommandData | Provide a structure for describing a command for a cooling actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| intensity | Describes the intensity value that a cooling actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Example

This example shows the description of an actuator command of cooling effect with the following semantics. This cooling actuator is commanded to perform the intensity of -2 celsius.

|  |
| --- |
| { "coolingCommandData": { "commandInfoBaseAttributes": {}, "intensity": { "unit": "celsius", "value": -2 } }} |

* + 1. Vibration actuator
			1. General

This sub-clause specifies the actuator command type which can generate a vibration effect.

* + - 1. Syntax

|  |
| --- |
| "vibrationCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" }, "frequency": { "$ref": "#/definitions/genericCommandType/genericFrequencyType" } }, "additionalProperties": false, "anyOf": [ { "required": [ "intensity" ] }, { "required": [ "frequency" ] } ] }, |

* + - 1. Semantics

Semantics of the vibrationCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| vibrationCommandData | Provide a structure for describing a command for a vibration actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| intensity | Describes the intensity value that a vibration actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |
| frequency | Describes the frequency value that a vibration actuator can generate. The frequency value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the frequency unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Example

This example shows the description of an actuator command of vibration effect with the following semantics. This vibration actuator is commanded to perform that the intensity is 40% of the maximum intensity and the frequency is 20% of the maximum frequency.

|  |
| --- |
| { "vibrationCommandData": { "commandInfoBaseAttributes": {}, "intensity": { "unit": "percentage", "value": 40 }, "frequency": { "unit": "percentage", "value": 20 } }} |

* + 1. Sprayer actuator
			1. General

This sub-clause specifies an actuator command type which can generate a spraying effect.

* + - 1. Syntax

|  |
| --- |
| "sprayerCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "sprayingType": { "$ref": "#/definitions/sprayingType" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false, "required": [ "sprayingType" ] }, |

* + - 1. Semantics

Semantics of the sprayerCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| sprayerCommandData | Provide a structure for describing a command for a sprayer actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| sprayingType | Describes the type of the sprayed material that shall be using the sprayingType. |
| intensity | Describes the intensity value that a vibration actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Examples

This example shows the description of an actuator command of the sprayer effect with the following semantics. This sprayer actuator is commanded to perform the material to be sprayed pure water, and the intensity shall be 45% of the maximum intensity.

|  |
| --- |
| { "sprayerCommandData": { "commandInfoBaseAttributes": {}, "sprayingType": "water", "intensity": { "unit": "percentage", "value": 45 } }} |

* + 1. Fog actuator
			1. General

This sub-clause specifies an actuator command type which can generate a fog effect.

* + - 1. Syntax

|  |
| --- |
| "fogCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false }, |

* + 1. Semantics

Semantics of the fogCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| fogCommandData | Provide a structure for describing a command for a fog actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| intensity | Describes the intensity value that a vibration actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + 1. Example

This example shows the description of an actuator command of fog effect with the following semantics. This fog actuator is commanded to perform the intensity 50% of the maximum intensity.

|  |
| --- |
| { "fogCommandData": { "commandInfoBaseAttributes": {}, "intensity": { "unit": "percentage", "value": 50 } }} |

* + 1. Wind actuator
			1. General

This sub-clause specifies an actuator command type which can generate a wind effect.

* + - 1. Syntax

|  |
| --- |
| "windCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "intensity": { "$ref": "#/definitions/genericCommandType/genericIntensityType" } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the windCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| windCommandData | Provide a structure for describing a command for a wind actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual device (actuator) is supposed to take. |
| intensity | Describes the intensity value that a vibration actuator can generate. The intensity value shall be set as either the percentage value with respect to the maximum value or the specific actuator value that the actuator can produce using genericIntensityType. If the intensity unit is "percentage", it is set to a percentage value and it is set to; otherwise, it is set to an absolute value.  |

* + - 1. Example

This example shows the description of an actuator command of wind effect with the following semantics. This wind actuator is commanded to perform the intensity 60% of the maximum intensity.

|  |
| --- |
| { "windCommandData": { "commandInfoBaseAttributes": {}, "intensity": { "unit": "percentage", "value": 60 } }} |

* + 1. Bubble actuator
			1. General

This sub-clause specifies the actuator command type which can generate a bubble effect.

* + - 1. Syntax

|  |
| --- |
| "bubbleCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the bubbleCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| bubbleCommandData | Tool for describing a bubble actuator command. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |

* + - 1. Example

This example shows the description of an actuator command of bubble effect with the following semantics. This bubble actuator is commanded to turn on to activate this actuator.

|  |
| --- |
| { "bubbleCommandData": { "commandInfoBaseAttributes": {} }} |

* + 1. Step motor actuator
			1. General

This sub-clause specifies the actuator command type which can control a step motor.

* + - 1. Syntax

|  |
| --- |
| "stepMotorCommandData": { "type": "object", "properties": { "commandInfoBaseAttributes": { "$ref": "#/definitions/commandInfoBaseAttributes" }, "speed": { "type": "integer", "minimum": 0 }, "steps": { "type": "integer", "minimum": 0 }, "orientation": { "type": "integer", "enum": [ 1, -1 ] } }, "additionalProperties": false }, |

* + - 1. Semantics

Semantics of the bubbleCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| stepMotorCommandData | Provide a structure for describing a command for a step motor actuator. |
| commandInfoBaseAttributes | Describes a group of attributes for the command data that each individual actuator is supposed to take. |
| speed | Describes the speed that the step motor actuator shall rotate in RPM (Rotations Per Minute) unit that the specific actuator can generate. |
| steps | Describes the number of steps that the step motor actuator shall rotate through a given angle.e.g., when the fixed step angle of the step motor is 5.625 degrees and the fixed speed variation ratio is 0.015625(1/64), the number of steps for one revolution is $\left(360÷5.265\right)×64=4,096$.For example, when the fixed step angle of the step motor is 5.625 degrees and the fixed speed variation ratio is 0.015625(1/64), setting steps to 2,048 means the stepper motor is commanded to perform the half revolution rotation. |
| orientation | Describes the orientation that the step motor actuator shall indicate the direction of rotation. If the orientation is 1, it means that the stepper motor rotates counterclockwise, and -1 means that it rotates clockwise. |
| orientation | Describes the orientation that the step motor actuator shall indicate the direction of rotation. If the orientation is 1, it means that the stepper motor rotates counterclockwise, and -1 means that it rotates clockwise. |

* + - 1. Example

This example shows the description of the actuator command of step motor with the following semantics. This step motor actuator is commanded to perform the 32 steps to rotate clockwise with the speed of 10 rpm.

|  |
| --- |
| { "stepMotorCommandData": { "commandInfoBaseAttributes": {}, "speed": 10, "steps": 32, "orientation": -1 }} |