|  |  |
| --- | --- |
| Project | **Standard for Actuator Interface for Cyber and Physical World**  <https://sagroups.ieee.org/2888/ **>** |
| Title | **Data Formats for Sight Related Actuator** |
| DCN | **2888-21-0023-00-0002** |
| Date Submitted | **June 15, 2021** |
| Source(s) | Yegi Lee [zxcasd312@naver.com](mailto:zxcasd312@naver.com) (Konkuk University)  Shin Kim [new.xin22@gmail.com](mailto:new.xin22@gmail.com) (Konkuk University)  Eunji Choi [c950707@gmail.com](mailto:c950707@gmail.com) (Konkuk University)  Kyoungro Yoon [yoonk@konkuk.ac.kr](mailto:yoonk@konkuk.ac.kr) (Konkuk University) |
| Re: |  |
| Abstract | This contribution proposes syntaxes, semantics, and examples for representing sight related actuator information in the physical world in a standardized data format. |
| 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 actuator(actuator) command types which can generate sight related effect. It contains syntaxes, semantics, and examples for representing sight related actuator information in the physical world in a standardized data format. Sight related actuators include light actuator and flash actuator.

2 Data formats for interfacing actuator command

* 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": {  "color": {  "$ref": "#/definitions/colorType"  },  "intensity": {  "type": "integer",  "minimum": 0  },  "intensityUnit": {  "$ref": "#/definitions/unitType"  }  }  }, |

* + 1. **Semantics**

The semantics of the lightCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| lightCommandData | Provide a structure for describing a command for a light actuator. |
| 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 of the lighting based on the intensityUnit in relation to the range of possible bright control. |
| intensityUnit | Specifies the intensity unit of the command value as a reference to a term that shall be using the unitType. |

* + 1. **Examples**

This example shows the description of the actuator command of light effect with the following semantics. This light actuator commands the intensity of 120 lux with the color “blue”.

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

* 1. **Flash actuator**
     1. **General**

This Subclause specifies a actuator command type which can generate a flash effect.

* + 1. **Syntax**

|  |
| --- |
| "flashCommandData": {  "type": "object",  "allOf": [{  "$ref": "#/definitions/lightCommandData"  }],  "properties": {  "frequency": {  "type": "integer",  "minumum": 0  },  "frequencyUnit": {  "$ref": "#/definitions/unitType"  }  }  }, |

* + 1. **Semantics**

Semantics of the flashCommandData:

|  |  |
| --- | --- |
| *Name* | *Definition* |
| flashCommandData | Provide a structure for describing a command for a flash actuator. |
| color | Describes the list of colors that the light actuator can provide either as a reference to a term that shall be using the colorType. |
| intensity | Describes the intensity of the lighting based on the intensityUnit in relation to the range of possible bright control. |
| intensityUnit | Specifies the intensity unit of the command value as a reference to a term that shall be using the unitType. |
| frequency | Describes the number of flickering based on the intensityUnit in relation to the range of frequency that the specific flash actuator can generate. |
| frequencyUnit | Specifies the frequency unit of the command value as a reference to a term that shall be using the unitType. |

* + 1. **Example**

This example shows the description of the actuator command of flash effect with the following semantics. This flash actuator commands the light intensity of 120 lux and frequency of 200 hz.

|  |
| --- |
| {  "commandInfoBaseAttributes": {},  "flashCommandData": {  "intensity": 120,  "intensityUnit": "lux",  "frequency": 200,  "frequencyUnit": "hz",  }  } |