|  |  |
| --- | --- |
| Project | **Standard for Actuator Interface for Cyber and Physical World**<https://sagroups.ieee.org/2888/ **>** |
| Title | **Application Programming Interfaces for Sight Related Actuators** |
| DCN | **2888-22-0019-00-0002** |
| 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 application programming interfaces for sight related actuators. |
| 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 application programming interfaces for sight related actuators.

Sight related actuators include:

* Light actuator
* Flash actuator

# API for individual actuators

Table 1 – Light API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Light() |
| *Default constructor.* |
|  |
| Light(String id) |
|  |
| Light(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| int | setLightComplexCommand(colorType color, int intensity) |
|  | *This function sets a command to designate a light color and control the intensity of the light actuator. This function shall include a color parameter defined by colorType from IEEE2888.2(Section 4.2.3) with the intensity parameter. If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |
| int | setLightColor(colorType color) |
|  | *This function sets a command to designate a light color of the light actuator. This function shall include a color parameter defined by colorType from IEEE2888.2(Section 4.2.3). If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |
| Int | setLightIntensity(int intensity) |
|  | *This function sets a command to control the intensity of brightness on the light actuator. This function shall include the intensity parameter. If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |

Table 2 – Flash API

|  |
| --- |
| Nested Classes |
| Modifier and Type | Method and Description |
|  |  |
| Constructor |
| Constructor and Description |
| Flash() |
| *Default constructor.* |
|  |
| Flash(String id) |
|  |
| Flash(String id, String serverIPAddress, integer serverPort) |
|  |
| Fields |
| Modifier and Type | Field and Description |
|  |  |
| Methods |
| Modifier and Type | Method and Description |
| int | setFlashComplexCommand(ColorType color, int intensity, int frequency) |
|  | *This function sets a command to designate a light color and control the intensity and frequency of the flash actuator. This function shall include a color parameter defined by colorType from IEEE2888.2(Section 4.2.3) with the intensity and frequency parameter. If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |
| int | setFlashColor(ColorType color) |
|  | *This function sets a command to designate a color of the flash actuator. This function shall include a color parameter defined by ColorType from IEEE2888.2(Section 4.2.3). If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |
| Int | setFlashIntensity(int intensity) |
|  | *This function sets a command to control the intensity of brightness on the flash actuator. This function shall include the intensity parameter. If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |
| Int | setFlashFrequency(int frequency) |
|  | *This function sets a command to control the frequency of the flash actuator. This function shall include the frequency parameter. If the command succeeds, this function returns 1; otherwise, it returns 0.* |
|  |  |