Interface Design Description (IDD) – ActuatorValue

**Abstract**

This document defines the template for the Interface Design Description of Arrowhead compliant Interfaces.

It provides a detailed description of how the ActuatorValue service is implemented to fetch and modify the Actuator status from the application system.

All Arrowhead Interface Designs should be specified using this template and stored on a common repository (available on the SVN server), in order to document and formalize the pilot demonstrators and the common Arrowhead framework.

Table of contents

[1. Interface Design Description Overview 2](#_Toc377455180)

[2. Interfaces 3](#_Toc377455181)

[3. Information Model 3](#_Toc377455182)

[4. References 3](#_Toc377455183)

[5. Revision history 4](#_Toc377455184)

[5.1. Amendments 4](#_Toc377455185)

[5.2. Quality Assurance 4](#_Toc377455186)

1. Interface Design Description Overview

This document describes the HTTP/{TLS}/JSON variant of the ActuatorValue service with REST interface. This allows for arbitrary Arrowhead Framework systems to fetch and modify the actuator data from each actuator present in the application system (fischertechnik Indexed line with two machining stations controlled with a Siemens S7-1500 PLC as asset via an OPC-UA server).

1. Service Interfaces

This section describes the interfaces that must be exposed by ActuatorValue services. In particular, the below subsection first names the HTTP method and path used to call the interface. The interface is expected to respond with HTTP status code 200 OK for all successful calls.

1. **GET {baseURL}/actautors**

* **Interface: GetActuators**
* **Output: Actuator JSON Objects**

Called to acquire the value read from all actuators from the assembly line.

Example of valid invocation:

GET /actuators HTTP/1.1

Accept: NA

Response: application/json

Example of valid response:

HTTP/1.1 200 OK

Content-Length:

Content-Type: application/json

[

{

"Id": "Q1",

"Value": "false",

"Description": "Q1 Motor slider 1 forward"

},

{

"Id": "Q2",

"Value": "false",

"Description": "Q2 Motor slider 1 backward"

},

{ ----------------

----------------

----------------

}

]

1. **GET {baseURL}/actuators/{actuatorId}**

* **Interface: GetActuator**
* **Output: Actuator JSON Object**

Called to acquire the actuator value read from the particular actuator.

Example of valid invocation:

GET /actuators/Q1 HTTP/1.1

Accept: String

Response: application/json

Example of valid response:

HTTP/1.1 200 OK

Content-Length:

Content-Type: application/json

{

"Id": "Q1",

"Value": "false",

"Description": "Q1 Motor slider 1 forward"

}

1. **PUT {baseURL}/actuators/{actuatorId}/{value}**

* **Interface: UpdateActuator**
* **Output: Actuator JSON Object**

Called to change the actuator value of the particular actuator.

Example of valid invocation:

PUT /actuators/Q1/true HTTP/1.1

Accept: Strings

Response: application/json

Example of valid response:

HTTP/1.1 200 OK

Content-Length:

Content-Type: application/json

{

"Id": "Q1",

"Value": "**true**",

"Description": "Q1 Motor slider 1 forward"

}

1. Information Model

Here, all the data objects that can be part of ActuatorValue service calls are listed in alphabetic order.

**Actuator**

JSON object with the following fields.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Description** | **Mandatory** | **Default** |
| Id | String | The Id of the individual actuator | True |  |
| Description | String | The OPC UA Variable name of the particular actuator | False |  |
| Value | Boolean | The value read by the particular actuator | False |  |

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2020-04-15 | 0.1 | First Draft | Aparajita Tripathy |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |