Service Description (SD) – ActuatorValue

**Abstract**

This document defines the template for the Service Description of Arrowhead compliant Services.

A Service Description provides an abstract description of what is needed for systems/devices/units/software to provide and/or consume a specific service.

SD’s for Application Services are created (specified) by the Pilots WP’s and by the common Arrowhead framework. The SD shall make it possible for an engineer with technical programming knowledge to achieve an Arrowhead compliant realization of a provider and/or consumer of description of how the service is implemented/realized by using the Communication Profile and the chosen technologies.

A Service Description (SD) is the Service in a specific technology. All systems/devices/units/software implementing an Interface Design Description which complies with this SD will be able to exchange information with each other.

Table of contents

1. Service Description Overview

2. Abstract Interfaces

3. Abstract Information Model

4. Non-Functional Requirement

5. Revision History

1. Service Description Overview

This document describes the ActuatorValue service offered by the “Fischertechnik AssemblyLine” system in the arrowhead cloud. This service allows consumer to get signal values read by the 10 actuators present in the fischertechnik factory. This also offers the consumer to write a signal value to the actuators.

1. Abstract Interfaces

This section lists the interfaces that must be exposed by the ActuatorValue service in alphabetical order. In particular, each subsection names an abstract interface, an input type, an output type and a set of possible exceptions, in that order. The input type is named inside parentheses, while the output type is preceded by a colon. Input and output types are only denoted when accepted or returned, respectively, by the interface in question. All abstract data types named in this section are defined in the Information model section.

**2.1 Interface GetActuators**

Called to get the value read by all 10 actuators.

**2.2 Interface GetActuator**

Called to get the value read by an actuator specified by an id parameter.

**2.3 Interface UpdateActuator**

Called to update the value of a specific actuator signal defined by an id parameter.

1. Abstract Information Model

Here, the data objects that can be part of ActautorValue service calls are listed in alphabetic order. Note that each subsection, which describes one type of object, begins with the struct or union keywords. The former is used to denote a collection of named fields, each with its own data type, while the latter is used to express that a value is allowed to be any one out of a number of listed variant types. For this service there is one object which is called:

**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. Non-functional Requirements

The service has none non-functional requirements.

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2020-04-25 | 0.1 | First draft | Aparajita Tripathy |

# Quality Assurance

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