GPIOController Service SD

1. GPIOController Overview

This document describes a low-level Arrowhead service for configuration of I/O pins, plis read and write capabilities. The document is also including the provides interfaces, functions and information model.

1. Abstract Interface

This simple Service provides control, read and write of General Purpose Input Output (GPIO) in digital values.

# getConfig

This method provides a GET request-response interface. If no request message, the method returns the configuration for all input/outputs. If a request message is provided, only the inputs/outputs in the request of type GPIOControlConfigRequest is returned in a GPIOControlConfigResponse.

# setConfig

This method provides a (POST) request-response. The request, of type GPIOControlConfigRequest, message contains configuration for all the pins that will be affected, and the response is a GPIOControlConfigResponse.

# getInputs

This method is a simple (GET) request-response. If no request message is provided, the method returns the configuration for all input/outputs. If a request message is provided (of type GPIOControlMessage), only the inputs/outputs in the request is returned in a GPIOControlMessage.

# setOutputs

This method provides a (POST) request-response. The request, of type GPIOControlMessage, message contains configuration for all the pins that will be affected, and the response is a GPIOControlMessage.

1. Abstract Information Model

The information model of the GPIOControlMessage uses the Sensor Markup Language [1] message ontology, which is a standard providing JSON and XML message formats to describe sensor readouts. The response payload contains the following information, as presented in Table 2.

Table 1 GPIOControlConfigRequest type description

|  |  |
| --- | --- |
| **Field** | **Description** |
| bn: String | Base Name, this field stores the name of the ArrowheadSystem |
| bt: Double | Base Time, this field contains the unix timestamp of the sensor readout |
| bu: String | Base Unit of the sensor readout |
| ver: Integer | Version identifier |
| e: List<MeasurementEntry> | List of measurements, containing the ArrowheadService, measured value and time |

Table 2 GPIOControlConfigResponse type description

|  |  |
| --- | --- |
| **Field** | **Description** |
| bn: String | Base Name, this field stores the name of the ArrowheadSystem |
| bt: Double | Base Time, this field contains the unix timestamp of the sensor readout |
| bu: String | Base Unit of the sensor readout |
| ver: Integer | Version identifier |
| e: List<MeasurementEntry> | List of measurements, containing the ArrowheadService, measured value and time |

Table 3 GPIOControlMessage type description

|  |  |
| --- | --- |
| **Field** | **Description** |
| bn: String | Base Name, this field stores the name of the ArrowheadSystem |
| bt: Double | Base Time, this field contains the unix timestamp of the sensor readout |
| bu: String | Base Unit of the sensor readout |
| ver: Integer | Version identifier |
| e: List<MeasurementEntry> | List of measurements, containing the ArrowheadService, measured value and time |

1. Non-functional Requirements

This Service is secured via the ArrowheadToken. Therefore, all Providers that implement this Service shall register this Service in the Service Registry with a special metadata: “security” = “token”.

1. References

[1] SensorML specifications: <https://tools.ietf.org/html/draft-jennings-senml-10>

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2019-02-11 | 1.0 | Initial | Jens Eliasson |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |
| 2 |  |  |  |