GPIOController Service IDD

**Abstract**

This document defines the REST-JSON-TLS-TOKEN interface of the GPIOController service.

1. GPIOController Service Overview

This document describes the GPIOController service, which is provided by a proof of concept Provider Application System of G4.0 with a REST interface.

1. Interfaces

Table 1 Function description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Path** | **Method** | **Input** | **Output** |
| getConfig | /config | HTTP GET | ArrowheadToken, GPIOControlConfigRequest | GPIOControlConfigResponse |
| setConfig | /config | HTTP POST | ArrowheadToken, GPIOControlConfigRequest | GPIOControlConfigResponse |
| getInputs | /pins | HTTP POST | ArrowheadToken, GPIOControlMessage | GPIOControlMessage |
| setOutputs | /pins | HTTP POST | ArrowheadToken, GPIOControlMessage | GPIOControlMessage |

This Service provides four interfaces, which is to configure GPIO pins and read and write their states. This information is provided in JSON, over HTTP(S).

When this Service is produced/offered over HTTPS, the metadata related to the security level has to be published in the Service Registry: **“security” = “token”. This is a mandatory Service metadata that must be associated when a System provides this.**

1. Information Model

The GPIO states are provided in SenML format [1]. The communication happens over HTTPS and the ArrowheadToken is implemented in the Service Provider resource when this Service is offered.

A direction is set using the bv or sv tags. A bv value of low or zero, or sv of “input” means input. A bv value of high or 1, or sv of “output” means output. A state value is set using the bv or sv tags. A bv value of low or zero, or sv of “low” means low. A bv value of high or 1, or sv of “high” means high.

The HTTPS requests are submitted generally to the following schema:

https://<providerAddress>:<port>/<serviceURI>/config?token=<token>&signature=<signature>

Where the <signature> and <token> fields are filled out with the ArrowheadToken that is received in the Orchestration response – or from the TokenGeneration service response (when consumed directly).

These fields cannot contain the URL escape characters. Therefore, if the token or signature contain those, they shall be replaced. For example, spaces (ASCII, hexa 20) are not allowed in the signature, they shall be replaced with ‘%2b’ in the path parameter.

An example response in JSON is presented in Fig. 2.

An example GPIOControlConfigResponse in JSON is presented in Fig. 1.

{

"bn": "\_insecureGPIOController.\_raspberrypi3-1.\_http.\_tcp.mydomain.local:8443",

"bt": 1506943344134,

"bu": "digital",

"e": [

{

"n": "gpio0",

"t": 0,

"bv": 0

},

{

"n": "gpio1",

"t": 0,

"sv": “output”

}

],

"ver": 1

}

**Message 1. Example GPIOControlConfigResponse, where the direction is 0, or low, or INPUT for gpio0, and string value “output” for gpio1.**

{

"bn": "\_insecureGPIOController.\_raspberrypi3-1.\_http.\_tcp.mydomain.local:8443",

"bt": 1506943344134,

"bu": "digital",

"e": [

{

"n": "gpio0",

"t": 0,

"bv": 0

}

],

"ver": 1

}

**Message 2. Example response, where the digital state is 0, or low.**

The fields are according to the SenML standard.

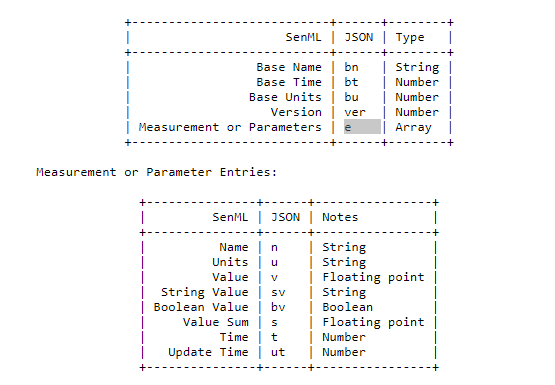


Figure 1: SenML encoding and types

1. References

[1] https://tools.ietf.org/html/draft-jennings-senml-07#page-7

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 |  |  |  |