Document Information

Info	Content
Document Creator	Manfred Siegl
Revision	1.0
Document Status	Draft
Date	Februar 5, 2015
Distribution	TUV
Keywords	G2021, Sedona Gbot
Selected Context	Communication protocol specification

Inhaltsverzeichnis

Preface....3
Protocol Specification....4
All Points Read Short....4
All Points Read Full....4
One Point Read....5
Point Write....5

Preface

The Software Defined Gateway G2021 consists of severl components. One of this components is the Interface-SW between Gbots and Sensors and Actors. The Sensors and Actors are elements connected to busses. On the G2021 there are a RS485-Modbus/RTU and CAN available. The drivers for Modbus and CAN are implemented in Sedona. The Interface-SW is implemented in Java.

This protocol specification defines the communication between the Interface-SW and the Sedona-Driver.

The information-flow is always initiated by the Interface-SW. The Sedona-Driver provides a http-Service. The Interface-SW requests information or action by sending an URI. The Sedona-Driver executes the requested action and responses some data. These data are formated in JSON.

The communication model is as follows:

1. The Interface-SW sends a http-request to the http-Service of the Sedona-Driver. The URL includes details of the request.
2. The Sedona-Driver trys to executes the request.
3. If the execution could be performed successfully, the Sedona-Driver responses with an JSON formated response-information.
4. If the execution could not be performed successfully, whatever the reason is, the Sedona-Driver responses with an JSON formated exception-information.

Each http-request is answered by one JSON formated information, either a response-information or an exception-information.

The answer is provided immediately after the request – within less then 100 Milliseconds.

If the answer includes the value of a sensor, it is of type float.

If the answer includes a selection, it is coded as number of type integer.

The Sedona-Driver will accept only one request concurrently. The response must be transfered, to accept a new request. Concurrent requests will not be processed.

Protocol Specification

In the following subsections, several requests and the corresponding responses are shown.

All Points Read Short

Verb	GET
URI	`rootpath/request/query?short`
Notes	Retrieve a list of all available points – only name, value and priority (optional)
Request	`rootpath/request/query?short`
Response	`{ "list" : [ { "name" : "point1", "value" : 1.27, "valid" : true }, { "name" : "point2", "value" : 0.0 , "valid" : true, "priority" : 8 }, { "name" : "point38", "value" : null, "valid" : false } ] }`
Exception	not applicable
Details	• If the value of the point is invalid, the value is set to `null` • `"priority"` is only included at point-type DO

All Points Read Full

Verb	GET
URI	`rootpath/request/query?full`
Notes	Retrieve a list of all available points – all available information
Request	`rootpath/request/query?full`
Response	`{ "list" : [ { "name" : "point1", "value" : 1.2, "valid" : true, "unit" : "&deg;" , "type" 1 }, { "name" : "point2", "value" : 0.0, "valid" : true, "unit" : "relais", "type" : 2, "priority" : 0 }, { "name" : "point38", "value" : null, "valid" : false, "unit" : "meter", "type" : 1 } ] }`
Exception	not applicable
Details	• If the value of the point is invalid, the value is set to `null` • The value of `"unit"` is html-encoded. • `"priority"` is included at point-type DO only • if the state of the point is notOK, the value is left empty • type: 1=DI, 2=DO, 3=AI, 4=AO

One Point Read

Verb	GET
URI	`rootpath/request/get?{point-name}`
Notes	Retrievs one point
Requ.Data	`rootpath/request/get?point21`
Response	`{ "list" : [ { "name" : "point21", "value" : 321.15, "valid" : true } ] }`
Exception	`{ "exception" : 1, "description" : "unknown point-name" }`
Details	• `"priority"` is included at point-type DO only • if the value of the point is invalid, the value is set to `null`

Point Write

Verb	GET
URI	`rootpath/request/set?{point-name}&{new-value}&{set-priority}`
Notes	Requests to set the output of *point-name* to be set to *new-value*. The set-operation has to be executed with priority *set-priority*
Request	`rootpath/request/set?port16&1.0&12`
Response	`{ "list" : [ { "name" : "point16", "value" : 0.0, "priority" : 5, "valid" : true } ] }`
Exception	`{ "exception" : 1, "description" : "unknown point-name" }` `{ "exception" : 2, "description" : "invalid point-type" }`
Details	• The point-name must exists. • The type of the point must be output - DO or AO. • If the upper conditions are not fullfilled, it leeds to an exception. • The set-priority should be in range 0..16. If set-priority is greater then 16, then 16 will be used. If set-priority is less then 0, then 1 will be used. • If set-priority is 0, the priority-condition will be switched off. • If a more prior condition exists, the set-operation is discarted and the former value is responded. • If the point is DO, new-value equal or greater then 0.5 is matched to ON and new-value below 0.5 is matched to OFF.