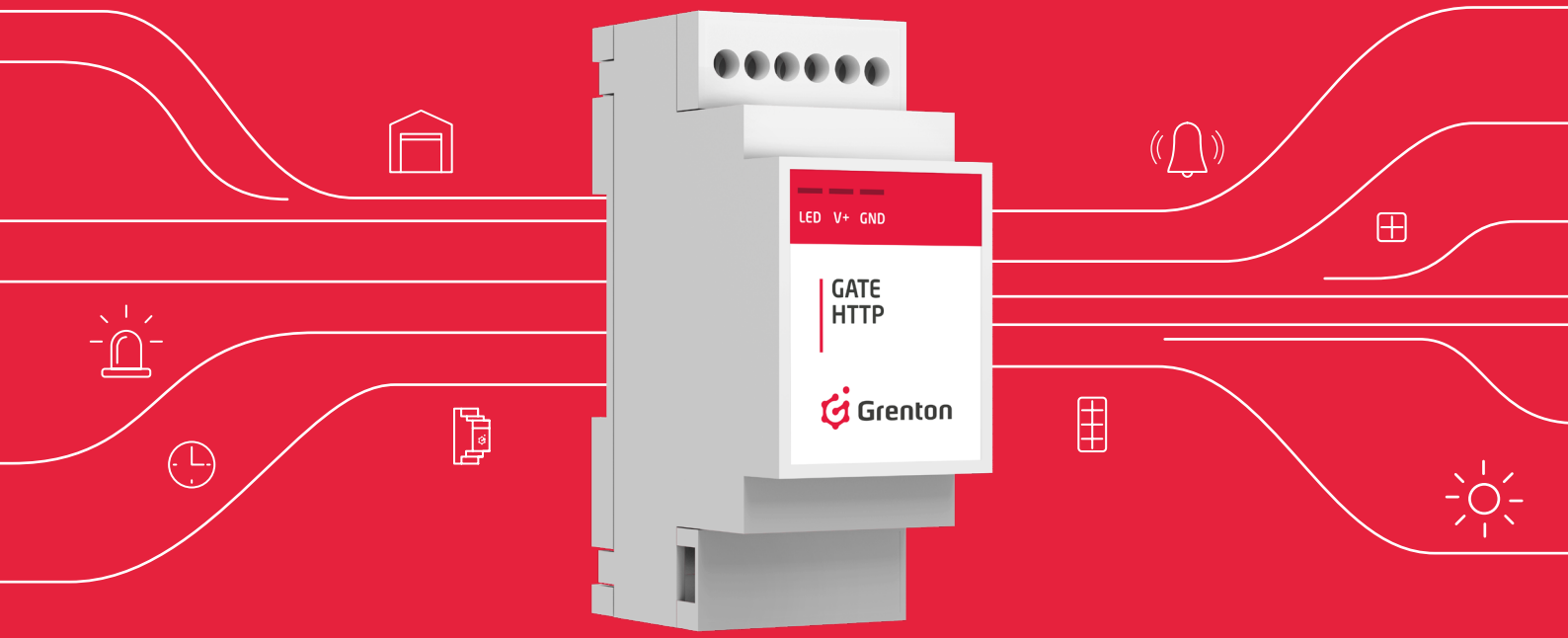
GRENTON GATE HTTP

INT-014-E-01

UNIVERSAL GATE MODULE FOR DIN RAIL ASSEMBLY



Module for integration of the Grenton system with external devices and systems. Gate HTTP enables integration with devices and services via HTTP protocol.

- supports both http client and server services (GET, POST)
- transmits information about modules subordinate to the CLU, which enables users to create complex integration logic in the Grenton system
- allows creation of event-based functionalities in the Grenton smart home environment

CONFIGURATION PARAMETERS

GATE_HTTP OBJECT

CHARACTERISTICS

Name	Description
UpTime	Time of device operation since last reset (in seconds)
UnixTime	Returns the current Unix time
FirmwareVersion	Gate software version
ClientReportInterval	Characteristics change report period

METHODS

Name	Description
SetDataTime	Sets date and time
SetClientReportInterval	Sets the characteristics change report period
StartConsole	Starts Lua console
StartConsoleOnReboot	Starts Lua console on next boot

EVENTS

Name	Description
OnInit	Event occurring once during device initialization

HTTPREQUEST OBJECT

ATTENTION! Characteristics described as the unchangeable contains only the responses. Its initial values must not be changed. Any operations with these variables needs to be done with the scripts and local variables.

CHARACTERISTICS

Name	Description
Host	Host address
Path	Query path
QueryStringParams	Query's parameters. \z means lack of parameters
Method	The type of method sent in the request ie. GET, POST
Timeout	Acceptable response timeout
RequestType	The type of content of the request being sent. Defines parameter content-type in the requests' header. Depending on chosen type, characteristic RequestBody is appropriately serialized:

	• 0 - None - unidentified. The content-type is not sent in the header. The RequestBody characteristic is not serialized. • 1 - Text - content-type: text/plain. The RequestBody characteristica is not serialized. • 2 - JSON - content-type: application/json. The RequestBody characteristic is serialized in JSON format. • 3 - XML - content-type: text/xml. The RequestBody characteristic is serialized in XML format. • 4 - FormData - content-type: application/x-www-form-urlencoded. The RequestBody characteristic is serialized to the table. • 5 - Other - content type (content-type) is different than built-in. The type may be defined in the header (RequestHeaders characteristic). The content is not serialized.
ResponseType	The type of expected answer. Defines parameter Accept in the request's header. Depending on chosen type, characteristic RequestBody is properly parsed to the table: • 0 - None - Accept parameter is not sent in the request's header. The answer (ResponseBody characteristic) is not parsed. • 1 - Text - Accept: text/plain. The answer (RequestBody characteristic) is not parsed. • 2 - JSON - Accept: application/json. The answer (RequestBody characteristic) is parsed in JSON format.v • 3 - XML - Accept: text/xml. The answer (RequestBody characteristic) is parsed in XML format. • 4 - FormData - Accept: application/x-www-form-urlencoded. The answer (RequestBody characteristic) is parsed. • 5 - Other - The header's parameter Accept is different than built-in. The type may be defined in the header (RequestHeaders characteristic).
RequestBody	The content of the message sent in the request. \z means lack of content
ResponseBody	The content of the message received after sending the request (characteristic used for reading in scripts - unchangable)
RequestHeaders	Additional HTTP request's headers
StatusCode	HTTP answer state

METHODS

Name	Description
SendRequest	Sends the request
AbortRequest	Breaks request's service
Clear	Deletes request's content
SetHost	Sets host's address
SetPath	Sets request's path
SetQueryStringParams	Sets query's parameters
SetMethod	Sets request's method
SetTimeout	Sets acceptable response timeout
SetResponseType	Sets the expected request's answer type
SetRequestHeaders	Sets additional HTTP request's headers
SetRequestBody	Sets the request's message content
SetRequestType	Sets the content type of the request being sent (content-type)

EVENTS

Name	Description
OnRequestSent	Event occuring when the request is sent
OnResponse	Event occuring when the answer is received

HTTPLISTNER OBJECT

ATTENTION! Characteristics described as the unchangeable contains only the responses. Its initial values must not be changed. Any operations with these variables needs to be done with the scripts and local variables.

CHARACTERISTICS

Name	Description
Path	Query path
Method	The type of method sent in the request ie. GET, POST
QueryStringParams	Returns HTTP query's parameters (characteristic used for reading in scripts - unchangable)
RequestType	The received request's type. Depending on the chosen type, the request's content (RequestBody characteristic) is appropriately parsed to the table: • 0 - None - The response is not parsed. • 1 - Text - The response is not parsed. • 2 - JSON - The response is parsed in JSON format. • 3 - XML - The response is parsed in XML format. • 4 - FormData - The response is parsed. • 5 - Other - The response is not parsed.
RequestBody	Returns HTTP request's content (characteristic using to read the value in scripts - unchangeable)
ResponseType	The type of sent request's answer. Defines parameter content-type in the response's header. Depending on chosen type, characteristic ResponseBody is properly serialized: • 0 - None - unidentified. The content-type is not sent in the header. The content is not serialized. • 1 - Text - content-type: text/plain. The RequestBody characteristic is not serialized. • 2 - JSON - content-type: application/json. The RequestBody characteristic is serialized in JSON format. • 3 - XML - content-type: text/xml. The RequestBody characteristic is serialized in XML format. • 4 - FormData - content-type: application/x-www-form-urlencoded. The RequestBody characteristic is serialized to the table.
ResponseBody	The content of HTTP response (characteristic used for reading in scripts)
StatusCode	HTTP response's state code. List of supported HTTP state codes: • 200 - OK • 201 - Created • 202 - Accepted • 204 - No Content

- 205 - Reset Content
- 400 - Bad Request
- 403 - Forbidden
- 404 - Not Found
- 405 - Method Not Allowed
- 406 - Not Acceptable
- 408 - Request Timeout
- 409 - Conflict
- 410 – Gone (Deleted)

METHODS

Name	Description
SendResponse	Sends the request's response
Clear	Deletes response's content
SetPath	Sets request's path
SetResponseType	Sets the expected request's answer type
SetResponseBody	Sets the response's content
SetStatusCode	Sets respone's state

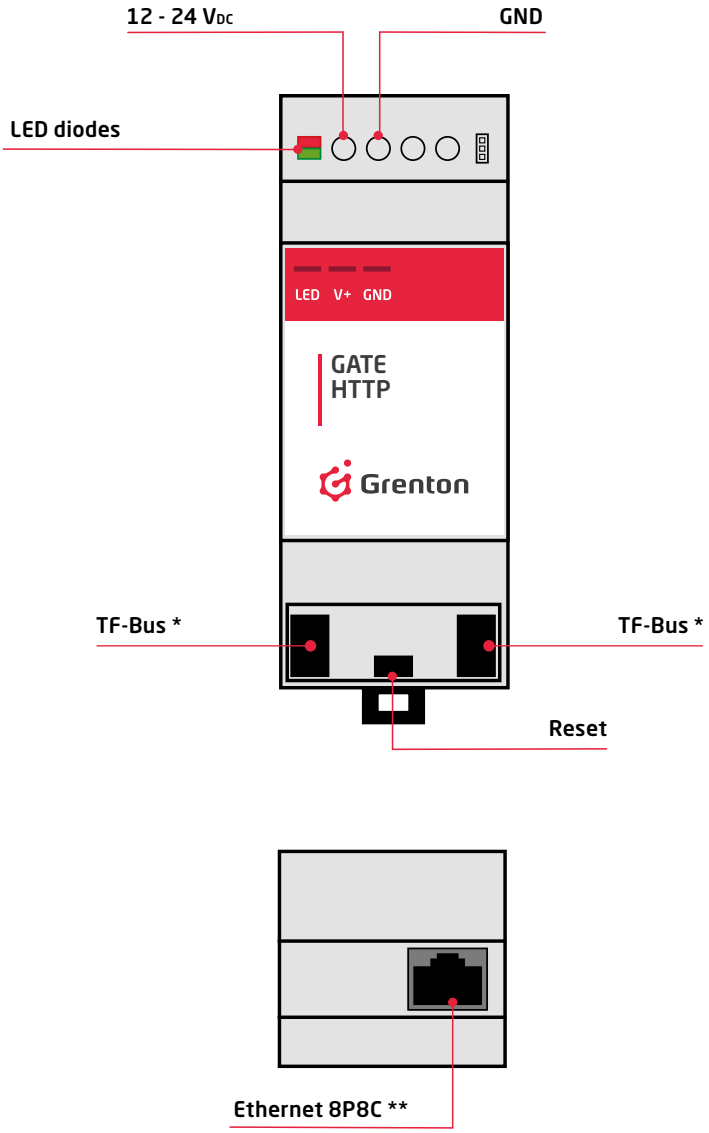
EVENTS

Name	Description
OnRequest	Event occuring when the request is received

TECHNICAL SPECIFICATION

power options	supply DC	12 - 24 V
	TF-Bus	5 V
max. current input for 12 V		80 mA
max. current input for 24 V		50 mA
weight		80 g
dimensions (H/W/D)		52/57/21 mm
operating temperature range		0 to +45°C

WIRING DIAGRAM



*) The device can be alternatively powered by means of the TF-Bus connector that provides 5 VDC power supply. However, it is recommended to use external 12 - 24 VDC supplies as it improves the stability of the system. The device works as a TF-Bus relay so it can be easily installed in the middle of other Grenton modules.

**) The GATE module communicates with the CLU unit via the Ethernet network/interface (8P8C Ethernet connector, so-called RJ-45).

LED - status indication:

- No supply
- Green diode blinks - system OK
- Error or no configuration