



Pushing Performance



People | Power | Partnership

# **HARTING ModbusTCP Gateway User Manual**

11. Edition 2018, 09/10

© HARTING IT Software Development, Espelkamp

All rights reserved, including those of the translation.

No part of this manual may be reproduced in any form (print, photocopy, microfilm or any other process), processed, duplicated or distributed by means of electronic systems without the written permission of HARTING Electric GmbH & Co. KG, Espelkamp

Subject to alterations without notice.

This ModbusTCP User Manual explains administration and configuration of the MICA. The following text refers to the Firmware Base Version V4.x and the Ha-VIS ModbusTCP Gateway App Version 2.1.0.

# Contents

1	Installation of the ModbusTCP Gateway	1
1.1	Basic container functionality	2
1.2	Network Settings	2
2	MODBUS/TCP	3
2.1	General	3
2.2	Modbus areas	3
2.3	Supported function codes	3
2.4	Modbus error codes	4
3	The ModbusTCP Gateway user interface	5
3.1	The section "Gateway Configuration"	6
3.2	The section "S0 Configuration" (only on MICA Energy)	7
3.3	The section "Firmware" (only on MICA Energy)	8
3.4	The section "Modbus Executor"	8
3.5	Error messages	10
3.6	The section "Devices"	12
3.7	The section "Reports"	14
4	The ModbusTCP Gateway modbus register interface	19
4.1	Function module configuration	19
4.2	S0 pulse counter contents	20
5	The ModbusTCP Gateway REST interface	20
5.1	Quick test using a browser	21
5.2	Reading data using the Chrome® "Postman" add on	22
5.3	Writing data using the Chrome® "Postman" add on	23
5.4	Reading data using the wget shell command within the same MICA	23
5.5	Reading data using Node-RED within the same MICA	24

### **Abstract**

The application "ModbusTCP Gateway" provides the functionality of a ModbusRTU to ModbusTCP Gateway for the MICA Energy with access to 8 build in S0 counters as well as every MICA with a RS485 interface.

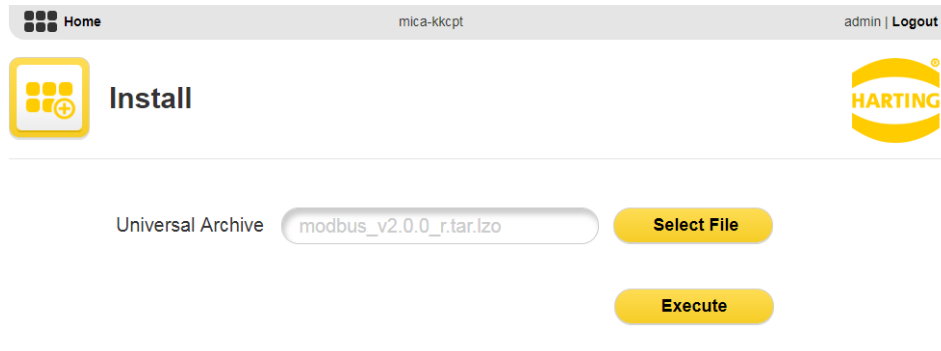
# 1 Installation of the ModbusTCP Gateway

Log in to the MICA with admin rights and click the "Install" icon.

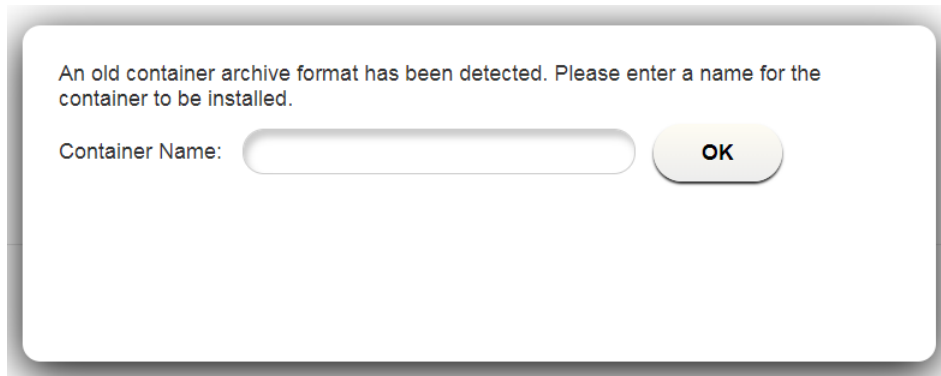


Install

On the next view, select the installation archive of the "ModbusTCP Gateway".



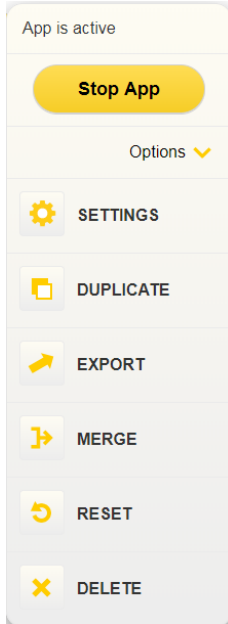
On the next view you can enter the name of the container to be installed. This name can then no longer be changed. Next click the OK button. The container will be installed.



As soon as the icon is visible you can start the container, the device is ready to receive Modbus/TCP commands sent to the IP address of the ModbusTCP Gateway container using the default port 2502. Dissenting to the Modbus/TCP default port of 502 the ModbusTCP Gateway uses the port 2502. It avoids access right problems on ports below 1000.

## 1.1 Basic container functionality

By executing an right click in the ModbusTCP Gateway icon a menu will appear containing the following functions.



1. **Settings** provides you useful information about the container and lets you configure the IPv4 and IPv6 settings.
2. **Duplicate** lets you duplicate the container on your MICA.
3. **Export** lets you export the container to your PC. All configurations you set will be kept.
4. **Merge** overwrites the reset point of the Container with its current configuration..
5. **Reset** resets the configuration of the ModbusTCP Gateway container to factory default.
6. **Delete** lets you delete the container.

## 1.2 Network Settings

In the containers context menu, click "SETTINGS" to configure the containers IPv4 setting and confirm by clicking the "Activate Settings" buttons. When done, go back to "Home" and click "Start App" in the containers context menu to start the container. The "ModbusTCP Gateway" is now ready to be connected to a device.

## 2 MODBUS/TCP

### 2.1 General

The MODBUS/TCP protocol is an extension of the serial MODBUS protocol optimized for the usage over a TCP/IP connection. It is a request/reply protocol and offers services specified by function codes (FC). The default port for MODBUS/TCP is 502.

Compared to the serial protocol a "MODBUS Application Protocol (MBAP)" header has been added to the protocol data unit. The CRC of the serial protocol is not necessary on a TCP transmission and therefore has been removed.

Table 1: Modbus/TCP message

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13..n
	MBAP Header						Protocol data unit							
	Transaction identifier		Protocol identifier		Length		Unit id.	Func. code	Reference number		Word count	Byte count	Data	



For further information refer to:

[http://www.modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf)

### 2.2 Modbus areas

The Modbus specification defines four Modbus areas.

Table 2: Modbus areas

Modbus areas	Data type	Access
Digital inputs (Discrete Inputs)	1 Bit	Read only
Digital outputs (Coils)	1 Bit	Read / write
Input registers	16 Bit	Read only
Output registers	16 Bit	Read / write

### 2.3 Supported function codes

The ModbusTCP Gateway app uses the 16 bit based "register" function codes only. The prefix "0x" marks numbers as hexadecimal.

Table 3: Supported function codes

Function code	Function name	Description
FC3 0x03	Read holding registers	reads multiple registers
FC4 0x04	Read input registers	reads multiple input registers
FC6 0x06	Write single register	writes a single register
FC16 0x10	Write multiple registers	writes multiple registers

## 2.4 Modbus error codes

The table below shows some general modbus error codes.



In this table, the modbus device is referred to as server, which a modbus environment is also often called a slave.

Table 4: Modbus error codes

Error code	Name	Description
0x01	Illegal function code	The function code received in the query is not an allowable action for the server.
0x02	Illegal data address	The data address received in the query is not an allowable address for the server.
0x03	Illegal data value	A value contained in the query data field is not an allowable value for the server.
0x04	Server failure	An unrecoverable error occurred while the server was attempting to perform the requested action.
0x05	Acknowledge	The server has accepted the request and is processing it, but a long duration of time will be required to do so.
0x06	Server Busy	The server is engaged in processing a long-duration program command.
0x07	Negative Acknowledge	The server cannot perform the program function received in the query.
0x08	Memory Parity Error	The server has detected a parity error in memory.
0x0A	Gateway problem	Gateway paths not available.
0x0B	Gateway problem	The targeted device failed to respond. The gateway generates this exception.

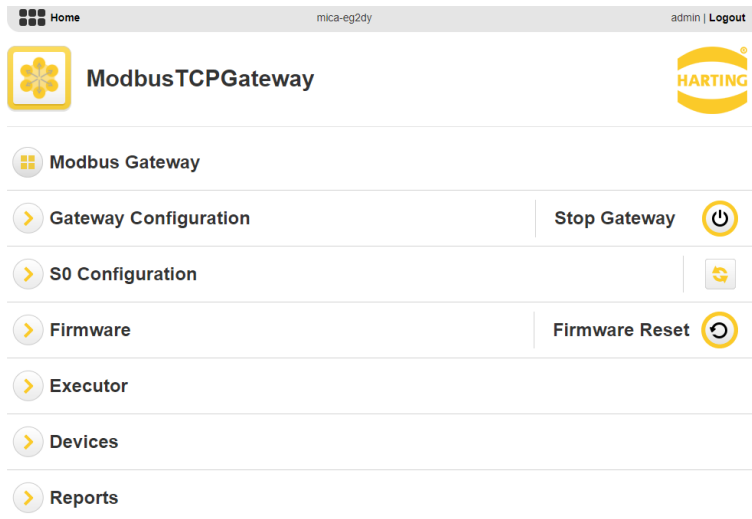


For further information about modbus errors refer to:  
[http://www.modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf)  
chapter 4.4.2.5 "MODBUS response building".



### 3 The ModbusTCP Gateway user interface

With a click on the *ModbusTCP Gateway* icon the user interface of the gateway appears. Here 6 different sections are available which provide needed information or let you configure the "ModbusTCP Gateway".



#### 3.1 The section "Gateway Configuration"

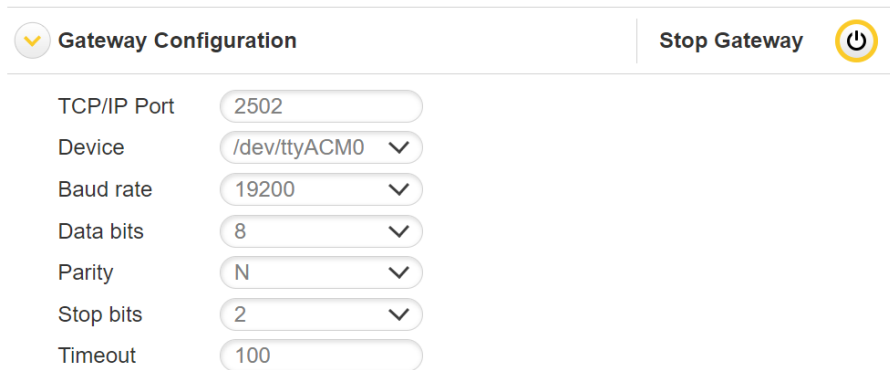
In the section "Gateway Configuration" you can set all parameters to operate the gateway. While using the IP address of the container you may here alter the factory set default port number (2502) according your needs. Furthermore all parameters which are used on the ModbusRTU (RS485) side can be set here. Make sure that all connected modbus devices work with identical settings. The ModbusTCP Gateway container can also be installed on MICA BASIC devices using an external connected USB device for RS485 communication. Therefore it is possible to choose different devices for the serial communication. There is no need to alter "ttyACM0" as long as you use the MICA Energy.



To alter the modbus settings, firstly you have to stop the gateway using the "Stop Gateway" button. It will immediately turn to "Start Gateway" which is afterwards used to start the gateway again.



**Attention!: Be sure that the Timeout is less than the timeout of your Modbus client!**





For a proper function of the RS485 serial line also consider the correct use of termination resistors. Further information you will find here:

[http://www.modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1.pdf](http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf)

## 3.2 The section "S0 Configuration" (only on MICA Energy)

The "S0 Configuration" section contains a list of all 8 S0 channels of the gateway "S0.1" to "S0.8". Each S0 channel appears its own ModbusRTU device providing its own slave id. Using the drop down boxes under "Slave ID" you may alter the factory defaults to your own requirements. Consider, that each slave id can only appear once within one ModbusRTU serial line.

**Modbus Gateway**

---

**Gateway Configuration** | **Stop Gateway**

---

**S0 Configuration** |

---

Channel	Slave ID	Pulse counter	Reset
S0.1	239	65536	
S0.2	240	0	
S0.3	241	0	
S0.4	242	255	
S0.5	243	0	
S0.6	244	0	
S0.7	245	0	
S0.8	246	0	

---

**Firmware** | **Firmware Reset**

The pulse counters will be kept in the MICA Energy and are protected against power loss. You may administrate them by entering values to the "Pulse counter" boxes or by resetting them using the "Reset" button.

The highest value of a single counter is 4294967295 - two 16bit words.



The reload button on the right side of "S0 Configuration" can be used to refresh the pulse counters.

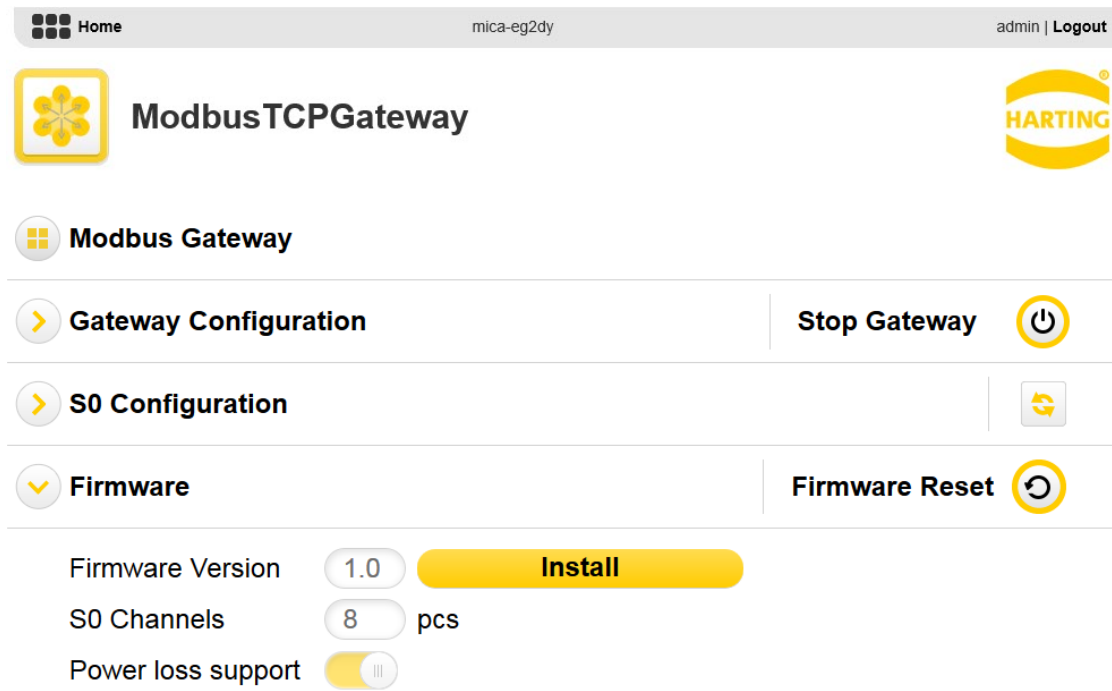
### 3.3 The section "Firmware" (only on MICA Energy)


The "Firmware" section contains some useful information about the RTU/S0 module.

**Firmware Version** displays the firmware version of the RTU/S0-module. If the yellow "Install" button is visible a new version is available and can be installed.

**S0 channels** displays the number of supported S0 channels.

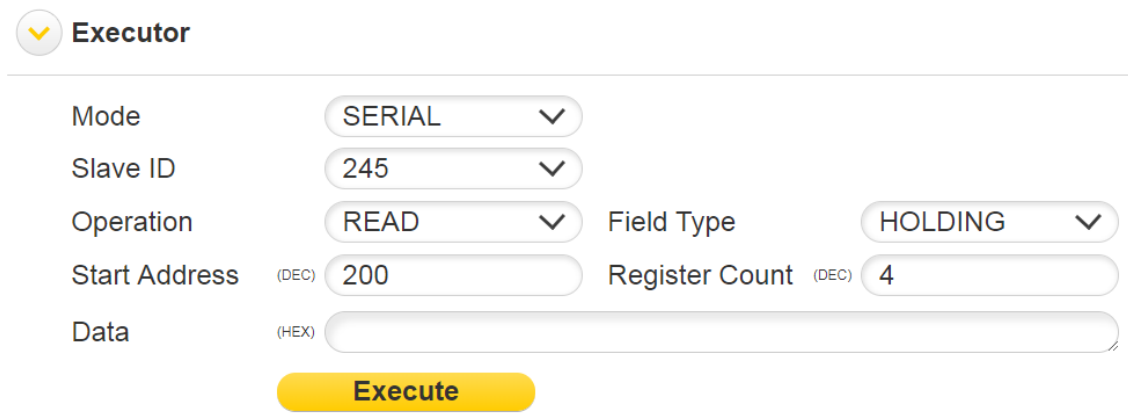
**Power loss support** indicates if the S0 counters will be secure saved even on a power loss.



 The "Firmware Reset" button on the RH side of "Firmware" can be used to reset the S0 configuration to factory defaults.

### 3.4 The section "Modbus Executor"

The "Modbus Executor" is a small tool to easily read and write data to devices which are connected to the current ModbusRTU line or to ModbusTCP devices.



To execute a Modbus command to RTU you have to provide 6 settings:

**Mode** Select item "SERIAL". It will only be available if the Gateway is enabled.

**Slave ID** Also called slave address. Select the ID of the desired slave device.

**Operation** Select either "READ" or "WRITE".


**Field Type** Select one of 4 different types. Together with the previously selected "Operation", the resulting Modbus function code is composed from that settings.

**Start Address** The Modbus register start address of the current used slave device.

**Register Count** The number of involved registers starting at the "Start Address" point.

Run the Modbus command by clicking the "Execute" button.

---

 **Executor**

---

Mode	<input type="text" value="TCP"/>		
Host	<input type="text" value="10.65.54.191"/>	TCP/IP Port	<input type="text" value="502"/>
Slave ID	<input type="text" value="1"/>		
Operation	<input type="text" value="READ"/>	Field Type	<input type="text" value="HOLDING"/>
Start Address	<small>(DEC)</small> <input type="text" value="200"/>	Register Count	<small>(DEC)</small> <input type="text" value="4"/>
Data	<small>(HEX)</small> <input type="text"/>		

**Execute**

To execute a Modbus command to TCP you have to provide 8 settings:

**Mode** Select item TCP. The item SERIAL will only be available if the Gateway is enabled.

**Host** Enter the host name of the ModbusTCP Device.

**TCP/IP Port** Enter the port of the ModbusTCP Device.

**Slave ID** Also called slave address. Select the ID of the desired slave device.

**Operation** Select either "READ" or "WRITE".

**Field Type** Select one of 4 different types. Together with the previously selected "Operation", the resulting modbus function code is composed from that settings.

**Start Address** The Modbus register start address of the current used slave device.

**Register Count** The number of involved registers starting at the "Start Address" point.

Run the Modbus command by clicking the "Execute" button.

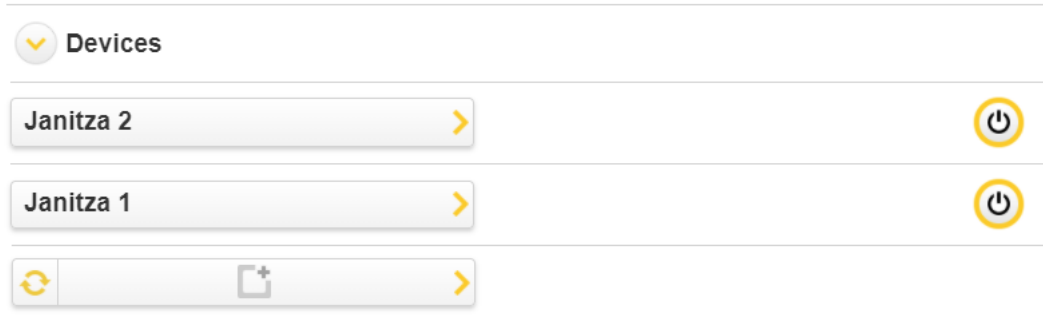
### 3.5 Error messages

In case of a wrong input (here too much digits for 2 registers), the write command responds with a "Error while writing" message.

Start Address	<input type="text" value="0"/>	Register Count	<input type="text" value="2"/>
Data	<input type="text" value="000000000D"/>		
<input type="button" value="Execute"/>			
<span style="color: red; font-weight: bold;">-</span> Error while writing <span style="float: right; color: red;">•</span>			

## 3.6 The section “Devices”

In this section you can add new devices you want to collect the data from.



By clicking the add button a window will appear containing the following fields:

1. **Name** is the displayed name of the device. The name will be also displayed in the section “Reports”.
2. **Period** is the sampling rate in ms.
3. **Properties** is for entering the type of the connection TCP/Serial and for choosing the corresponding property Host/Port. slave = slave ID.

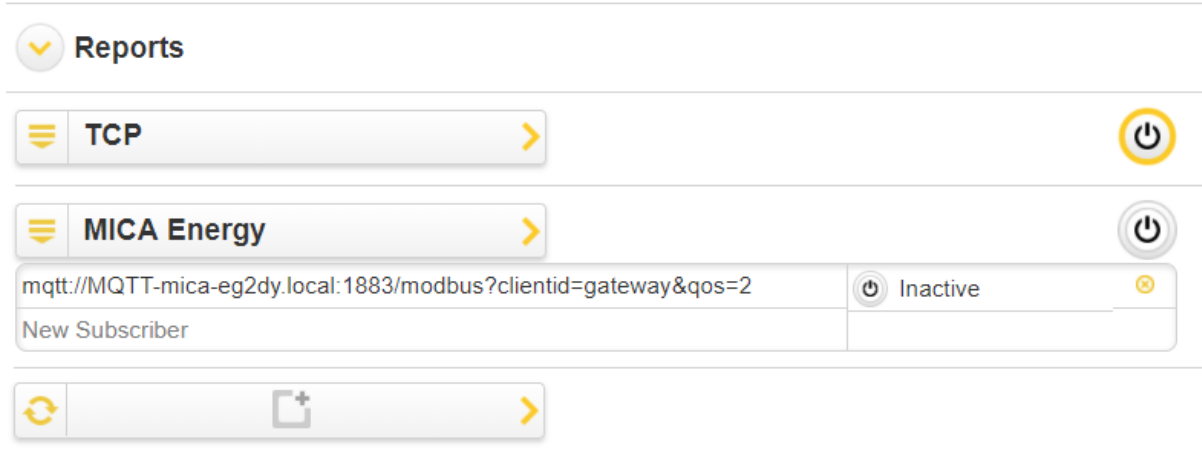
By clicking the “Fields” button a field box will appear. When you click in the box a new window appear where you can enter the following values.

1. **Name** is the name of the value.
2. **Access** is the function code.
3. **Address** is the start register.
4. **Length** is the number of the register.
5. **Type** is the representation type of the data.

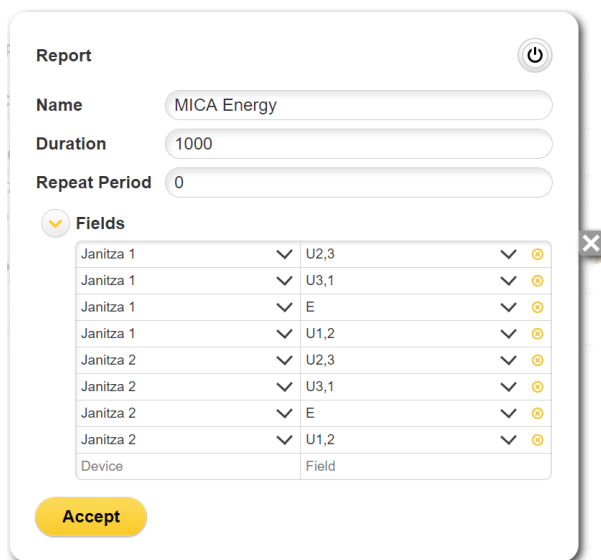
If you have entered all needed information click on the “Accept” button to add your device to the list.

### 3.7 The section "Reports"

Here you can configure reports you want to get from the devices you added in the section "Devices".



If you want to add a new report click on the add field. A window will appear where you can enter the following information:



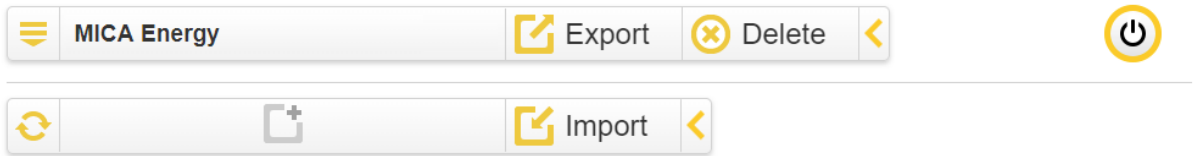
1. **Name** is the displayed name of the report.
2. **Duration** is the duration of the reports in ms.
3. **Repeat Period** is the point the next report starts. As long as the repeat period is smaller as the duration the start of the next report is immediately after the last.
4. **Fields** is for choosing your device you added in the section "Devices" and the value you want to observe.

If you have entered all needed information click on the "Accept" button to add the report.

#### 3.7.1 Import/ Export Reports and Devices

If you hover with the cursor over a device or a report, a menu appear on the left side of it. Here you can delete a device/report and you can "Export" it to save the current configuration or to create a template for same devices so you have only to change the slaveID.

## ▼ Reports




For import a device or report hover with the cursor over the add field. The "Import" field will appear. Click on that field and choose the device/report you want to import and click open. The device/report will be imported.

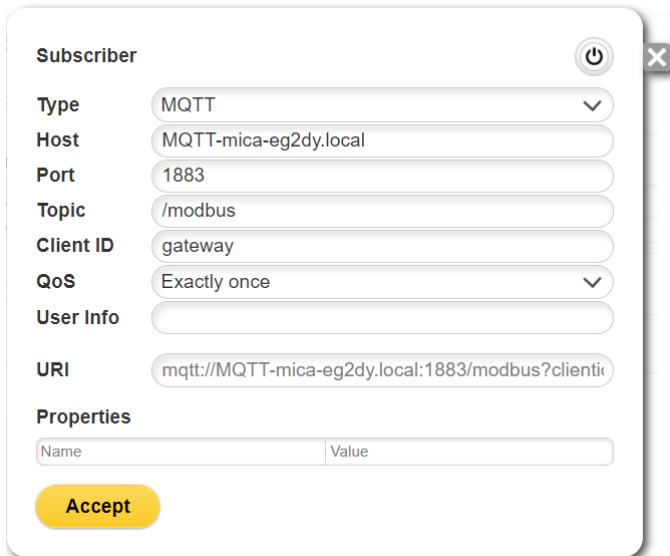
You can import several device/report specifications at once. If a specification with the same name already exists, a number is added to the name. The devices and reports are added in deactivated state.



### 3.7.2 Subscriber

When you have added the report you can click on  and a drop-down menu will appear where you can add a new subscriber endpoint. The supported types of endpoints are HTTP, HTTPS, MQTT, MQTTS, TCP and UDP.

For adding an new subscriber click in the field “New Subscriber” and a window will appear where you can choose the type of the endpoint and enter all needed information. You find an example for a subscriber endpoint in the picture below.



### 3.7.3 Report Example

If you have configured your “Devices”, “Reports” and your subscriber endpoint, you will get a report after the time which was specified by you.

A Report is a JSON structure containing the following fields:

```

Report
applicationID : String;
reportName : String;
date : String;
totalMilliseconds : Integer;
devices : Device[];
    
```

The fields of the Reports are specified in the following table:

Field	Type	Description
applicationID	String	The ID of the application.
reportName	String	The name of the report which is set in the section Reports.
date	String	The the time stamp of the report.
totalMilliseconds	Integer	The duration of the report.
device	Device []	The all monitored devices.

You find an example of a report in the field below:

```
{
  "applicationId": "ModbusGateway",
  "reportName": "MICA Energy",
  "date": "2017-11-18T21:13:04.843",
  "totalMilliseconds": 1000,
  "initiation": "REQUESTED",
  "initiator": null,
  "termination": "DURATION",
  "terminator": null,
  "devices": [
    {
      "name": "Janitza 1",
      "fields": [
        {
          "name": "U2,3",
          "value": 162.5906,
          "date": "2017-11-18T21:13:03.959"
        },
        {
          "name": "U3,1",
          "value": 162.60217,
          "date": "2017-11-18T21:13:03.974"
        },
        {
          "name": "E",
          "value": 162.60217,
          "date": "2017-11-18T21:13:03.990"
        },
        {
          "name": "U1,2",
          "value": 0.35030943,
          "date": "2017-11-18T21:13:04.005"
        }
      ]
    }
  ]
}
```



## 4 The ModbusTCP Gateway modbus register interface

The settings described in the user interface are also accessible via standard modbus registers.

### 4.1 Function module configuration

The following described registers provide access to the build in S0 pulse counters on the MICA Energy function module (FM) and the module itself by means of modbus holding registers. Use the modbus function FC3 - "Read Holding Registers" to read this registers. If possible, use the modbus function FC16 - "Write Multiple Registers" to write to this registers. The prefix "0x" marks numbers as hexadecimal.

Table 5: Function Module, Device ID: 247

Register address	Length (Word)	Register type	Access	Description
0x0000	1	Holding	R/W	<b>Function Module info</b>  <b>Read</b> MSB == 0x00: Power loss unsupported MSB == 0x01: Power loss supported LSB: number of S0 channels (8)  <b>Write</b> A value of 0xAA55 restores the factory default configuration. Default value: 0x0108.
0x0001	1	Holding	R/W	<b>S0 channel 1 - device ID</b> Range: 1 - 246, Default value: 239.
0x0002	1	Holding	R/W	<b>S0 channel 2 - device ID</b> Range: 1 - 246, Default value: 240.
0x0003	1	Holding	R/W	<b>S0 channel 3 - device ID</b> Range: 1 - 246, Default value: 241.
0x0004	1	Holding	R/W	<b>S0 channel 4 - device ID</b> Range: 1 - 246, Default value: 242.
0x0005	1	Holding	R/W	<b>S0 channel 5 - device ID</b> Range: 1 - 246, Default value: 243.
0x0006	1	Holding	R/W	<b>S0 channel 6 - device ID</b> Range: 1 - 246, Default value: 244.
0x0007	1	Holding	R/W	<b>S0 channel 7 - device ID</b> Range: 1 - 246, Default value: 245.
0x0008	1	Holding	R/W	<b>S0 channel 8 - device ID</b> Range: 1 - 246, Default value: 246.
0x8000	1	Holding	R	<b>Firmware version</b> MSB: major LSB: minor

## 4.2 S0 pulse counter contents

The device Id's of the 8 S0 channels can be freely assigned in a range between 1 and 246. Each address of the complete modbus RTU line can be used once only. Using the modbus address 0x0000 of each device, the S0 pulse counter can be read or written to.

Table 6: S0 counter contents

Register address	Length (Word)	Register type	Access	Description
0x0000	2	Holding	R/W	<b>S0 pulse counter</b> of a given channel unsigned int (32 bit) 0x00000000 - 0xFFFFFFFF

## 5 The ModbusTCP Gateway REST interface

The "ModbusTCP Gateway" container provides a REST interface" which allows gathering information from the gateway and its connected slaves without the need of using the ModbusTCP protocol. For an automatic access to the gateway it is essential to provide authorization data normally included in the request headers of the commands. The needed encrypted content can be copied e.g. from the Chrome® add on "Postman" as shown in the following code snippets. In case of a newly started browser, you will be requested to enter your credentials prior to the first execution of the REST command.



Make sure using the IP address of the MICA base when calling the REST interface from a client outside the MICA.

If you call the interface from a container within the same device you can use the non secured http-protocol to save access time.

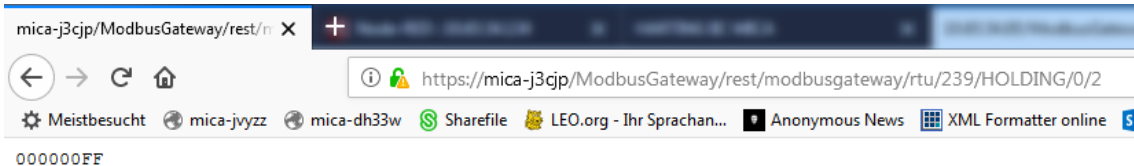
## 5.1 Quick test using a browser

### 5.1.1 RTU

By means of a simple GET command you can retrieve information from the device. Enter the following command in a browsers command-line:

```
https://mica-xyz/<containerName>/rest/modbusgateway/rtu/239/HOLDING/0/2
```

You will see:



In the command-line you will find all the details you also have to specify in the "Modbus Executor". The device address of the MICA base `https://mica-xyz/` is followed by the container name and the path to the rtu section of the gateway `ModbusGateway/rest/modbusgateway/rtu/`, followed by:

- The Slave ID **239**/, followed by:
- The Field Type **HOLDING**/, followed by:
- The Start Address **0**/, followed by:
- The Register count **2**, followed by:

The Operation mode (READ) is automatically selected by means of the GET command.

### 5.1.2 TCP

Using a similar GET command you can retrieve information from the device via TCP. Enter the following command in a browsers command-line:

```
https://mica-xyz/<containerName>/rest/modbusgateway/tcp/10.65.54.191/502/239/HOLDING/0/2
```

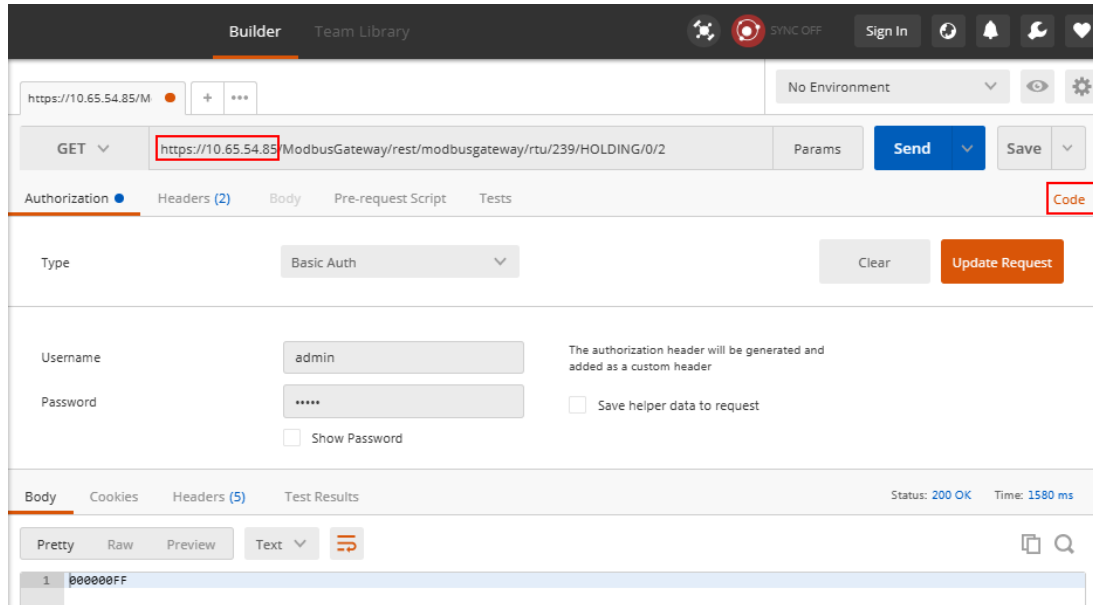
In the command-line you will find all the details you also have to specify in the "Executor". The device address of the MICA base `https://mica-xyz/` is followed by the container name and the path to the tcp section of the gateway `ModbusGateway/rest/modbusgateway/tcp/`, followed by:

- The hostname **10.65.54.191** followed by:
- The port **502** followed by:
- The Slave ID **239**/, followed by:
- The Field Type **HOLDING**/, followed by:
- The Start Address **0**/, followed by:
- The Register count **2**, followed by:

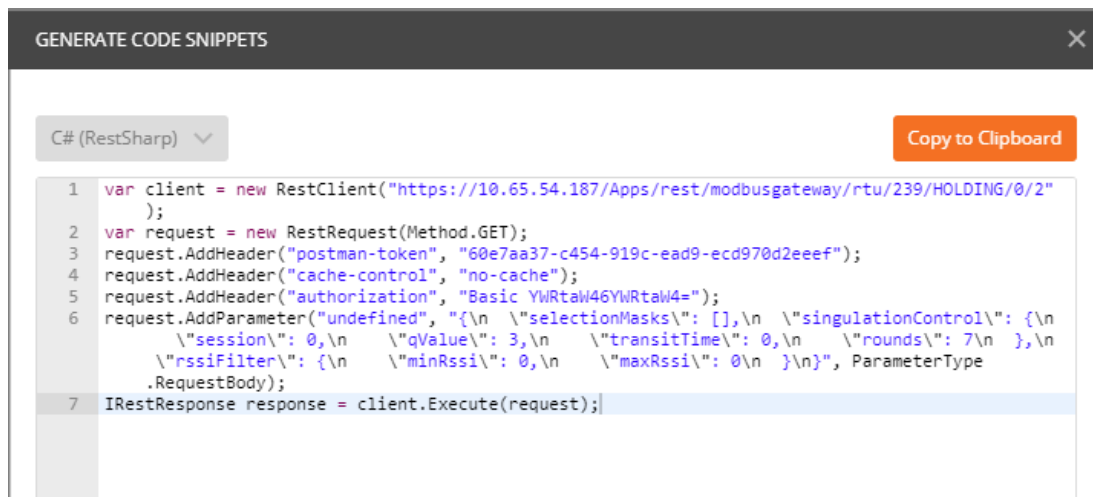
The Operation mode (READ) is automatically selected by means of the GET command.

## 5.2 Reading data using the Chrome® "Postman" add on

Software like the Chrome® add on "Postman" can also be used to send GET commands. It gives a better overview on how the commands behave and provides more details.



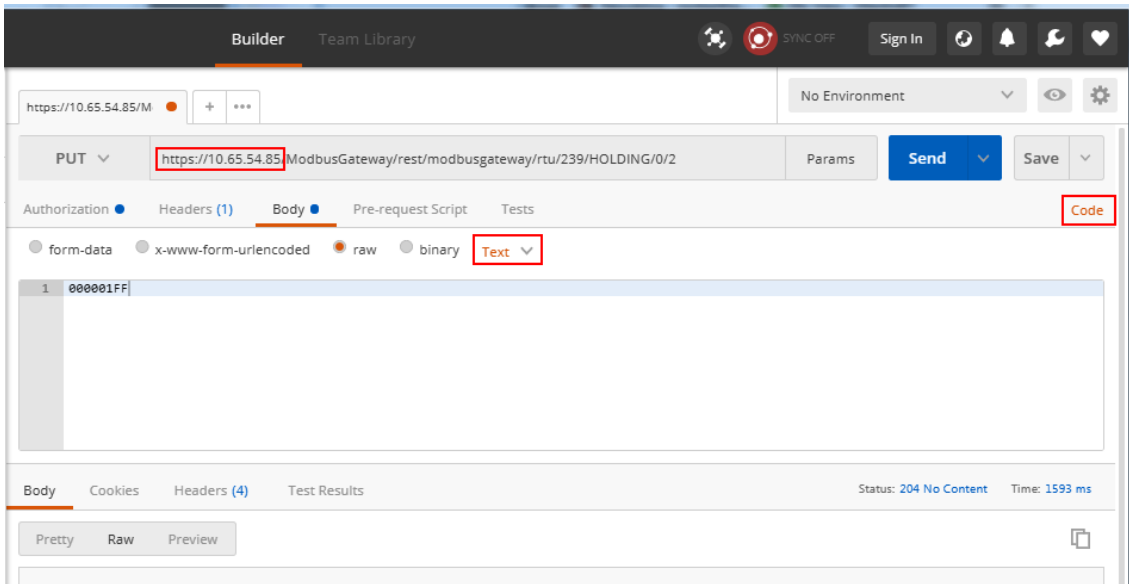
On clicking to "Code", Postman generates code snippets for using the REST commands in many other programming languages, here C#:



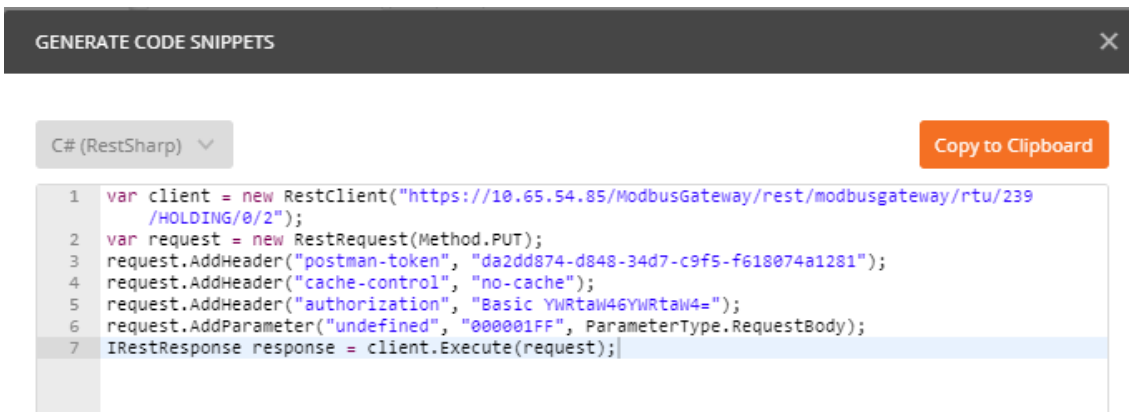
### 5.3 Writing data using the Chrome® "Postman" add on

By means of a PUT command you may write data to the desired slave-registers. To test it, use again the Chrome® add on "Postman" and enter e.g.

`https://mica-xyz/<containerName>/rest/modbusgateway/rtu/239/HOLDING/0/2`



Here too, a generated code snippet for C#:



### 5.4 Reading data using the wget shell command within the same MICA

As mentioned at the beginning of that chapter, you may also use the REST interface between two different containers of the same device. Using the shell command "wget" is a very useful approach for a quick test of the REST interface.

The sample script below accepts on parameter for the IP address, stores the result in a variable and prints it on the console.

```

#!/bin/sh
IP=$1
printf "Call: %s\n" $IP
RESULT="`wget -qO- http://admin:admin@$IP/rest/modbusgateway/rtu/239/HOLDING/0/2`"
printf "Result: 0x%s\n" $RESULT
    
```

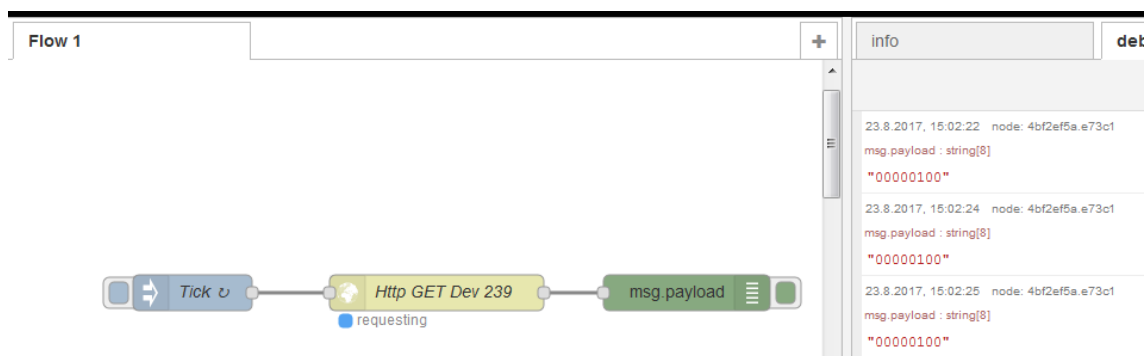
Using an ssh software like "Putty" log yourself in to another container as the Apps container and create a new file called "get.sh". Make it executable and call it using the syntax: ./get.sh 10.65.54.146.


The output should look like:

```
10.65.54.180 - PuTTY
root@Depot-mica-j3cjp:~# ./get.sh 10.65.54.146
Call: 10.65.54.146
Result: 0x00000100
root@Depot-mica-j3cjp:~#
```

## 5.5 Reading data using Node-RED within the same MICA

Also the Node-RED container available at <https://www.harting-mica.com/> can be easily used to establish connections to the Gateway Apps REST interface. Start Node-RED and use the inject, the http request and the debug node to compose a scenario sending permanent Http-GET commands to the Gateway.



Here you may download the sample flow:  [httpget.json](#) (Acrobat only.)