CAREL – Confidential



**IoT simulator  
 How to use it**

*Gateway Middle End*

rev. See revision table

DRAFT

Index

[Index 2](#_Toc39497952)

[Revisione 3](#_Toc39497953)

[1 Instructions 4](#_Toc39497954)

[1.1 Prerequisites 4](#_Toc39497955)

[1.1 Connect to the server 4](#_Toc39497956)

Revisione

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Rev. date | Author | Note |
| *0.01* | *30/04/2020* | *A.Bilato* | 1° draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Instructions

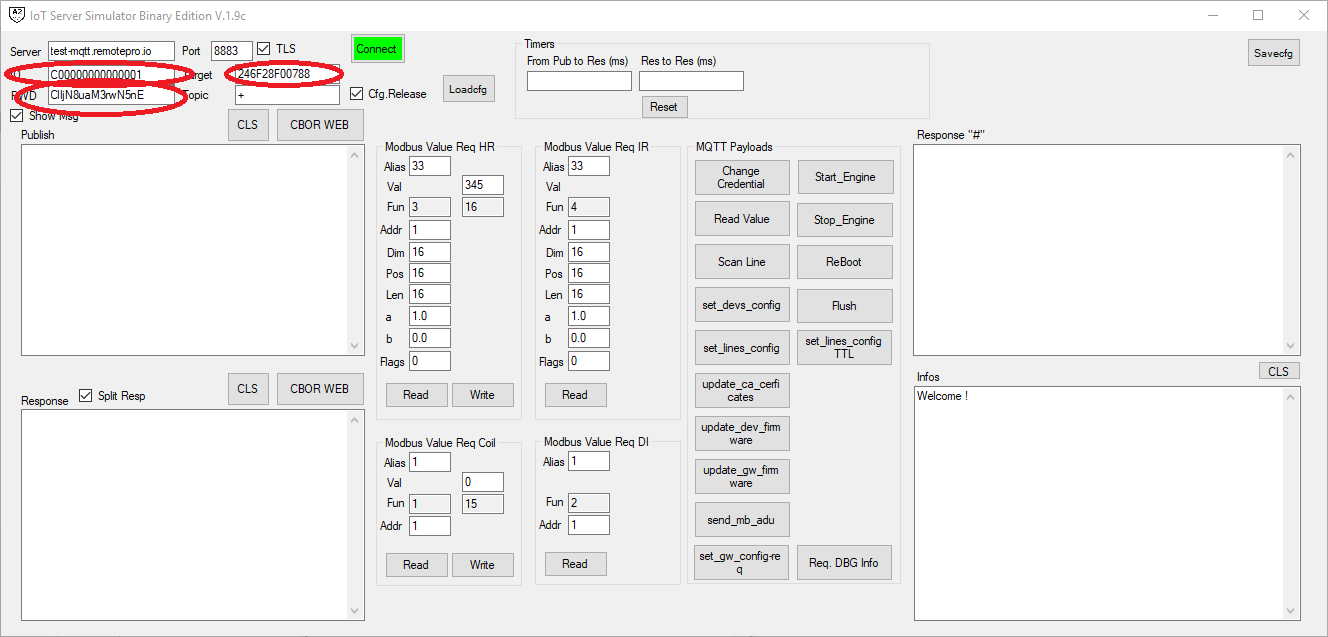
Here below some brief instruction about the use of the program   
“IoT Server Simulator Binary Edition V.x.x.x” to test some function of the Gateway Middle End (GME).

**WARNING!**

**This software is for CAREL INTERNAL USE ONLY, do not share it outside CAREL.**

* 1. Prerequisites
* This program require the Miscosoft .NET Framework V.xxx please install it before launch the program.
* Unzip the zzzzzz.zip file in a suitable folder and double click on MqttClientSimulatorBinary.exe
  1. Connect to the server

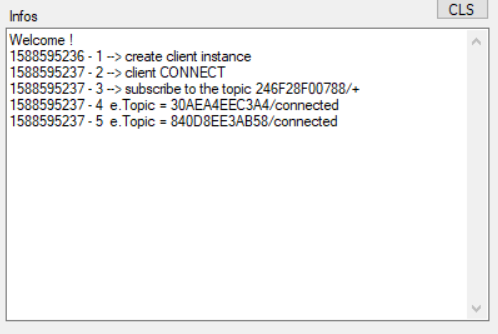
Immediately after the launch this screen is displayed

  
Fig.1

Follow these steps (see red circles):

* Power off the GME under test
* Set the “ID” to C00000000000001
* Set the “PWD” to ClIjN8uaM3rwN5nE
* Set the “Target” to the MAC address (if you use the WiFi model the IMEI for the 2G) of your GME.
* Press the “Connect” button
* Power on the GME

If always, work right in a few seconds you will see in the “Infos” box



Something like this take a look that the subscribed topic point to the MAC of the device under test

and in a few seconds the communications of the GME   
  
  
  
you will see also all the communications of other devices in the “Infos” box.



In the “Response” box, you will see only the communication from the selected GME to the cloud and viceversa.

So, in the “Response” box you will see something like this  
{"ver": 257

"t": 1588596515

"sta": 1}

{"ver": 257

"pn": "GTW000MWT0"

"hwv": 256

"fwv": 256

"btm": 1588596515

"bau": 19200

"mqv": h'030101'

"did": 0

"gid": h'00000000000000000000000000000000'

"crc": 54456

"cid": 0}

{"ver": 257

"t": 1588596518

"upt": 3

"fme": 130868

"est": 3

"sgn": -45}

{"ver": 257

"t": 1588596515

"sta": 1}

The 1st is the “hello” the 2nd the status message.

Try now a very simple command to check the system, press the button “ReBoot”,   
in the “Response” box will appear

{"ver": 257

"rto": "reboot"

"cmd": 2

"cid": 65536}

{"ver": 257

"rto": "reboot"

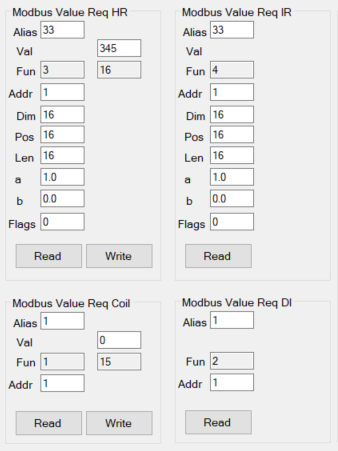
"cmd": 2

"res": 0}

The blue message is the message sent from the simulator to the GME, the 2nd is the GME feedback to the command.  
If everything work right, the GME will reboot.

* 1. Command to remotely use Modbus

The GME normally do a polling loop and read some data from the attached device but is possible to remotely read or write any value of the device attached to the GME, to do that use the panel in Fig.2

 Fig.2

Please leave unchanged the “Alias”, “a”, “b” “Flags” fields are useful for the cloud.  
The rest are quite simple to understand “Addr” is the address of registry/coil/DI/IR you want to R/W.

When you press “Read” or “Write” a message will be sent to the GME and the corresponding action will be done.  
The GME response will be appear on the “Response” box.