

User Manual and Test Guide

Modbus Master Function

1. FUNCTIONS	3
2. SIMPLIFIED MODBUS MASTER TEST	4
2.1. CELLULAR NETWORK PRODUCTS TEST	4
2.2. ETHERNET AND WI-FI PRODUCTS TEST	12
3. COMPLETE MODBUS MASTER TEST	20
3.1. MODBUS MASTER POLLING FUNCTION	20

1. FUNCTIONS

The Modbus Master function is used for the product to initiate Modbus polling. It queries the Modbus Slave device status and sends it to the server.

There is simplified and complete version of modbus master function, the difference is as following.

- Simplified version supports max 10 query command, complete version does not have any limit
- Simplified version supports JSON or HEX format upload, complete version support JSON
- Simplified version supports any query protocol (not only modbus, can be set any command), complete version support modbus only
- All HF IOT Device support the simplified version, but only part IOT Device support complete version.

The following products support Modbus Master complete version.

Ethernet + Wi-Fi + serial port: HF2211, HF2221, HF9610

Ethernet + Wi-Fi + IO Control: HF6208

Ethernet + Wi-Fi + 4G + serial port: HF2421, HF2421G, HF9624

Ethernet +4G+ serial port: EG46

以太网物联	Wi-Fi物联	PLC物联网关	4G物联
超级网口 [Eport-E20-PIN] [Eport-E20] [Eport-E30] Linux超级网口 [Eport Pro-EP20-PIN] [Eport Pro-EP20] [Eport Pro-EP40] 以太网串口服务器 [HF5111S] [HF5111A] [HF5111B] 以太网多串口服务器 [HF5122] [HF5142A] [HF5142B]	Wi-Fi DTU模块 [Wport-W20] [Wport-W10] Wi-Fi串口服务器 [HF2211S] [HF2211] Wi-Fi多串口服务器 [HF2221] wifi 路由器 (导轨) [HF8104W]	4G+以太网+Wi-Fi+串口 [HF9624] 4G+以太网+Wi-Fi [HF9606] 以太网+Wi-Fi [HF9606W] 以太网+Wi-Fi+串口 [HF9610]	4G 串口服务器 [HF2411] 4G+WIFI+GPS 串口服务器 [HF2421G] 4G+WIFI串口服务器 [HF2421] 4G路由器(导轨式) [HF8102] [HF8104]
小精灵物联	轨精灵物联	远程IO控制	2G/NB-IOT物联
4G+以太网 [Elfin-EG46] 4G_LTE [Elfin-EG41] GPRS [Elfin-EG11&Elfin-EG10] Wi-Fi [Elfin-EW10&Elfin-EW11] 以太网 [Elfin-EE10&Elfin-EE11] EMC 神盾 [Elfin-EEAX0] 配网精灵 [Elfin-ECA10]	4G_LTE [Protoss-PG41] GPRS [Protoss-PG11] WiFi+以太网 [Protoss-PW21] Wi-Fi [Protoss-PW11] 以太网 [Protoss-PE11]	4G蜂窝网IO控制 [HF6408] Wi-Fi型IO控制 [HF6208] 以太网型IO控制 [HF6508]	NB-IOT串口服务器 [HF2611] GPRS串口服务器 [HF2111] [HF2111A]

For basic products usage and config procedure, see operation guide for detail, this doc won't guide the basic usage.

Related tools can be downloaded from the following address:

http://www.hi-flying.com/index.php?route=download/category&path=1_4

2. SIMPLIFIED MODBUS MASTER TEST

2.1. Cellular Network Products Test

Cellular network products use AT command to config, the detailed product type is as following, this case take EG11 for example.

Ethernet IOT	Wi-Fi IOT	PLC IOT	4G IOT
FreeRTOS Embedded Network Device [Eport-E20-PIN] [Eport-E20] [Eport-E30]	Wi-Fi Serial Module [Wport-W20] [Wport-W10]	4G+Ethernet+Wi-Fi+serial [HF9624]	4G Serial Server [HF2411]
Linux Embedded Network Device [Eport Pro-EP20-PIN] [Eport Pro-EP20] [Eport Pro-EP40]	Wi-Fi Serial Server [HF2211S] [HF2211]	4G+Ethernet+Wi-Fi [HF9606]	4G+WIFI+GPS Serial Server Device [HF2421G]
Ethernet Serial Server [HF5111S] [HF5111A] [HF5111B]	Multiple Port Wi-Fi Serial Server [HF2221]	Ethernet+Wi-Fi [HF9606W]	4G+WIFI Serial Server [HF2421]
Multiple Port Ethernet Serial Server [HF5122] [HF5142A] [HF5142B]	Wifi router (rail) [HF8104W]	Ethernet+Wi-Fi+serial [HF9610]	Rail 4G Router [HF8102] [HF8104]

Elfin IOT	Protoss IOT	IO Control	2G/NB-IOT
4G+Ethernet [Elfin-EG46]	4G_LTE [Protoss-PG41]	4G IO [HF6408]	NB-IOT Serial Server [HF2611]
4G_LTE [Elfin-EG41]	GPRS [Protoss-PG11]	Wi-Fi IO [HF6208]	GPRS Serial Server [HF2111] [HF2111A]
GPRS [Elfin-EG11&Elfin-EG10]	WiFi&Ethernet [Protoss-PW21]	Ethernet IO [HF6508]	
Wi-Fi [Elfin-EW10&Elfin-EW11]	Wi-Fi [Protoss-PW11]		
Ethernet [Elfin-EE10&Elfin-EE11]	Ethernet [Protoss-PE11]		
EMC Protect [Elfin-EEAX0]			
Elfin config [Elfin-ECA10]			

Upgrade firmware to 2.0.6 version.

I.O.T Service Serial

Close COM Auto GetIn CMD Quit CMD Reload Restart Upgrade Update Script Read Device Write Device Batch Set

PC Serial Para
COM: COM4 Baudrate: 115200 Data Bits: 8 Parity: NONE Stop Bits: 1

DTU Para

UART Para
UART No: UART Baudrate: 115200 Data Bits: 8 Parity: NONE Stop Bits: 1
Flow Control: Disable UART Protocol: NONE
HeartBeat Time: 0 HeartBeat Serial: ...

SOCKET
SOCKET Name: A Protocol: OFF Rout: ...
Server Addr: ... Server Port: 0
Connect Mode: Always Burst Time: 300
HeartBeat Time: 0 HeartBeat Serial: ...
Regist Mode: Disable Regist Code: ...
Data Tag: Disable Data Tag Code: ...
Security: Disable Security Key: ...

SIM Para
IMEI: 860344041066789 ICCID: 89860439101880606970
Status: Connected RSSI: 22 Refresh

Others
ModuleSN: 344041066789 Welcome: Gport-EG11 HostName: Gport-EG11
APN: APN User: APN Passw...
Latitude: 0.000000 Longitude: 0.000000 Detail

SEND:AT+IMEI
RCV:+ok=860344041066789
SEND:AT+ICCID
RCV:+ok=89860439101880606970
SEND:AT+HEART=1
RCV:+ok=1,0,IOTWORKSHOP
SEND:AT+UARTTM=1
RCV:+ok=1,0
SEND:AT+MODBUS=1
RCV:+ok=1,off
SEND:AT+UART=1
RCV:+ok=1,115200,8,1,NONE,NFC
SEND:AT+NETP=A
RCV:+ok=A,1,off
SEND:AT+NETP=B
RCV:+ok=B,1,off
SEND:AT+NETP=C
RCV:+ok=C,1,UDP,0.0.0.0,28987
SEND:AT+HEART=C
RCV:+ok=C,0
SEND:AT+NREGSND=C
RCV:+ok=C,link
SEND:AT+NREGDT=C
RCV:+ok=C,
SEND:AT+NREGEN=C
RCV:+ok=C,off
SEND:AT+NETPID=C
RCV:+ok=C,
SEND:AT+NETPIDEN=C
RCV:+ok=C,off
SEND:AT+NETPLK=C
RCV:+ok=C,off
SEND:AT+VER
RCV:+ok=2.0.6(2020-04-02 11:00)
SEND:AT+HOST
RCV:+ok=Gport-EG11
SEND:AT+WEL
RCV:+ok=Gport-EG11

EG11 2.0.6(2020-04-02 11:00)

Clear Send

Update HIS script (ModbusMasterScript)

I.O.T Service Serial

Close COM Auto GetIn CMD Quit CMD Reload Restart Upgrade Update Script Read Device Write Device Batch Set

PC Serial Para
COM: COM4 Baudrate: 115200 Data Bits: 8 Parity: NONE Stop Bits: 1

DTU Para

UART Para
UART No: Baudrate: 115200 Data Bits: 8 Parity: NONE Stop Bits: 1
Flow Control: Disable UART Protocol: NONE
HeartBeat Time: HeartBeat Serial: ...

SOCKET
SOCKET Name: A Protocol: OFF Rout: ...
Server Addr: ... Server Port: 0
Connect Mode: Always Burst Time: 300
HeartBeat Time: HeartBeat Serial: ...
Regist Mode: Disable Regist Code: ...
Data Tag: Enable Data Tag Code: ...
Security: Disable Security Key: ...

SIM Para
IMEI: ICCID: ...
Status: RSSI: ...

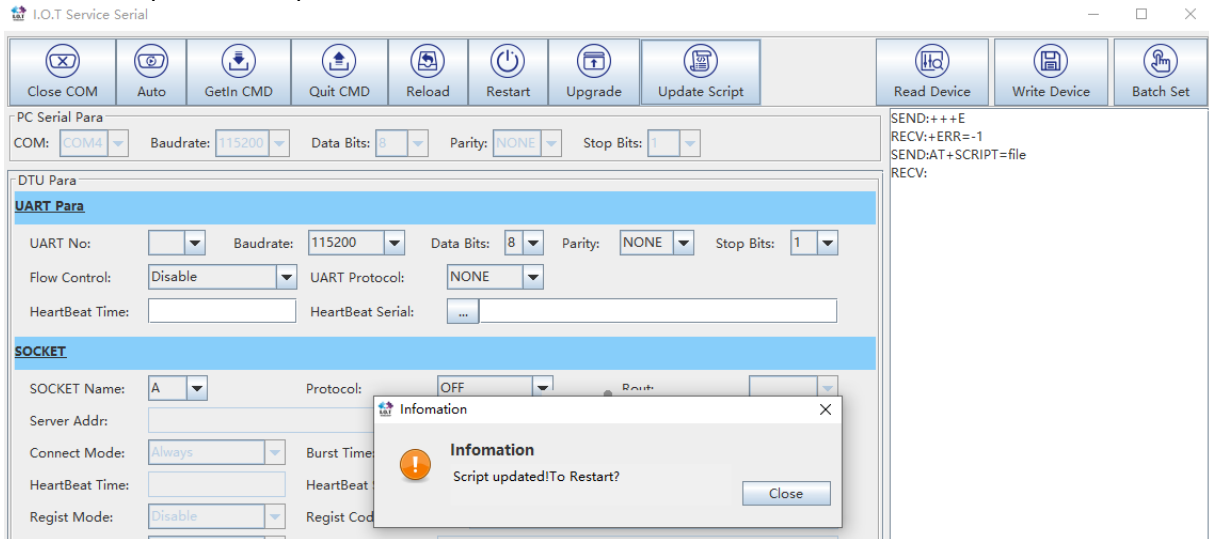
Others
ModuleSN: Welcome: HostName: ...
APN: 3gnet APN User: APN Passw...
Latitude: Longitude: Detail

SEND:+++
RCV:+ERR=-1

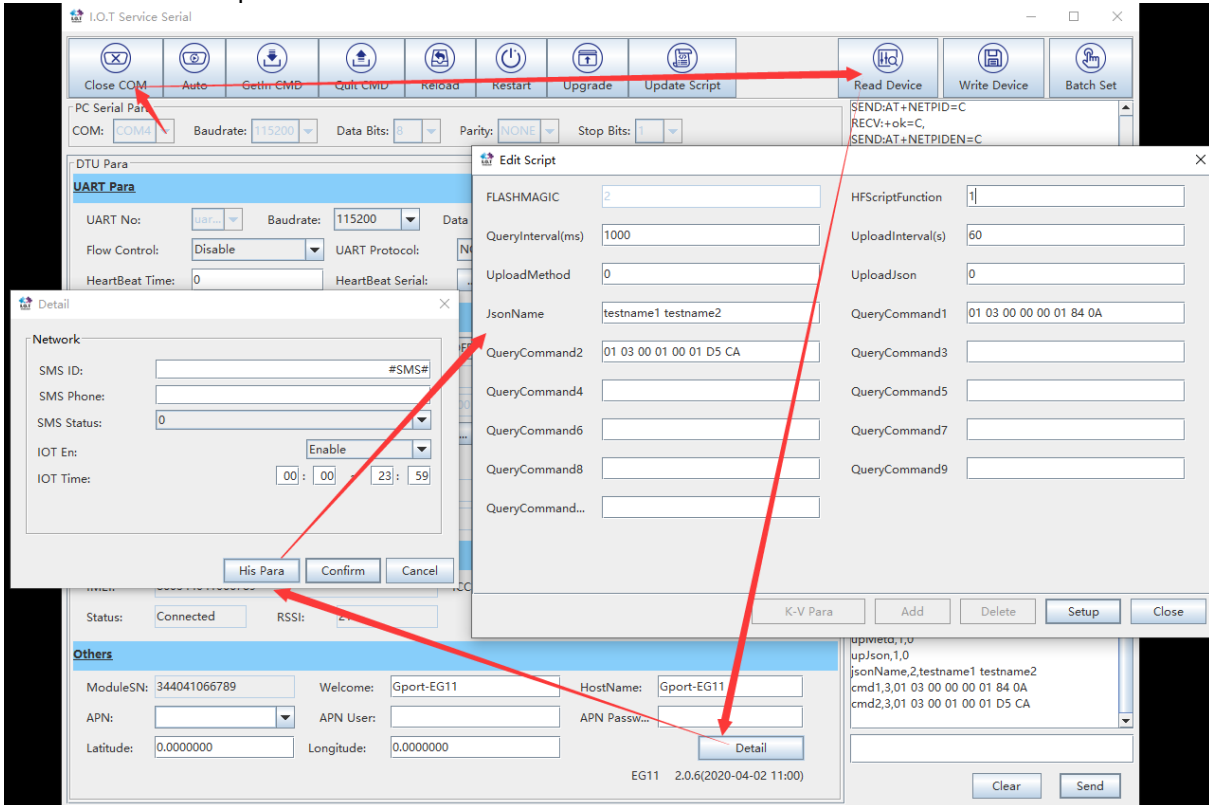
Select
查找(I): Desktop
学习资料 mqt测试命令.txt
云智易 备注.txt
019 - 快捷方式 EW_smart_meter_script_reference.txt 工具升级HF2111A卡住
FTP.txt 路由器.txt
HTTP请求.txt 写MAC地址.txt
ModbusMasterScript-20200305.txt
文件名(N): ModbusMasterScript-20200305.txt
文件类型(T): *.txt
Select 取消

Clear Send

Restart products if update success.



Check HIS Script content



FLASHMAGIC: HIS Para initial flag, if the flag is not the same as previous, then all the HIS para will be initialized to the script file content. For example, when first update this script, all the HIS para will be initialized, when second time update script, the HIS para won't be initialized, keep the previous.

HFScripction: Query function: enable or disable, 1: Enable Query. 0: Disable Query.

QueryInterval(ms): The interval time of each query command, range from 200ms to 60000ms

UploadInterval(s): The interval time of several query commands, range from 10sec to 86400 sec. For example,

Query1->QueryInterval->Query2...->Query10->UploadInterval->Query1->QueryInterval->....

UploadMethod: 0 or 1, 0: Upload immediately when receive response of each Query Command packet, 1: Upload after receive all the Query Command response packet.

UploadJson: 0 or 1, 0: Upload the response packet. 1: transform the response packet to JSON and upload.

Upload Json: Only valid when Upload Method is 1, Name number should be same as query command number. maximum 10characters for each command.

- UploadMethod set to 0, Upload Method set to 1 example.

Upload 1: //value is the response packet in ASCII format

```
{
  "key": "Name1",
  "value": "xxxxxxx"
}
```

Upload 2:

```
{
  "key": "Name2",
  "value": "xxxxxxx"
}
```

- UploadMethod set to 1, Upload Method set to 1 example.

Upload

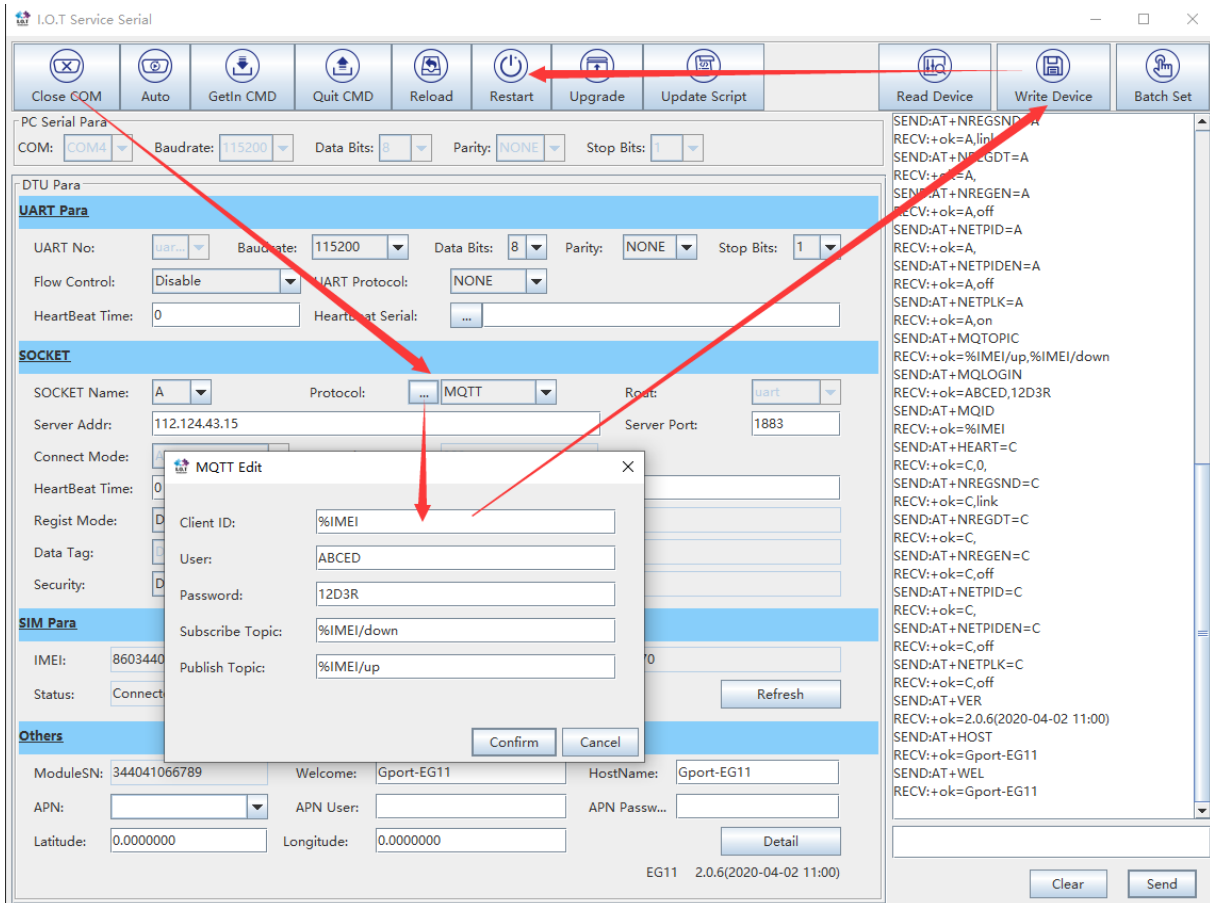
```
{
  "Packets": [{
    "key": "Name1",
    "value": "xxxxxxx"
  }, {
    "key": "Name2",
    "value": "xxxxxxx"
  }, {
    "key": "Name3",
    "value": "xxxxxxx"
  }
}]
```

QueryCommand: Hex format, max 10 commands, 80 bytes max for each command.

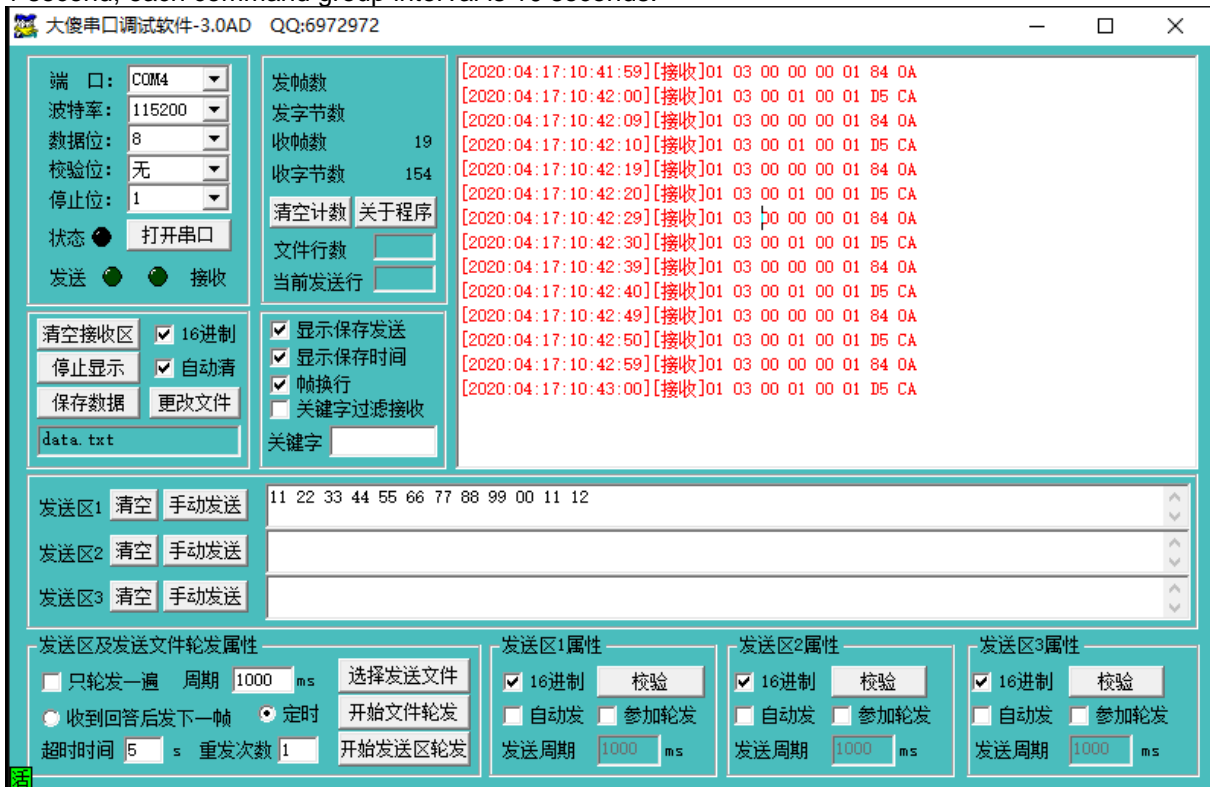
Edit Script
 ×

FLASHMAGIC	<input type="text" value="2"/>	HFScriptFunction	<input type="text" value="1"/>
QueryInterval(ms)	<input type="text" value="1000"/>	UploadInterval(s)	<input type="text" value="10"/>
UploadMethod	<input type="text" value="0"/>	UploadJson	<input type="text" value="0"/>
JsonName	<input type="text" value="testname1 testname2"/>	QueryCommand1	<input type="text" value="01 03 00 00 00 01 84 0A"/>
QueryCommand2	<input type="text" value="01 03 00 01 00 01 D5 CA"/>	QueryCommand3	<input type="text"/>
QueryCommand4	<input type="text"/>	QueryCommand5	<input type="text"/>
QueryCommand6	<input type="text"/>	QueryCommand7	<input type="text"/>
QueryCommand8	<input type="text"/>	QueryCommand9	<input type="text"/>
QueryCommand...	<input type="text"/>		

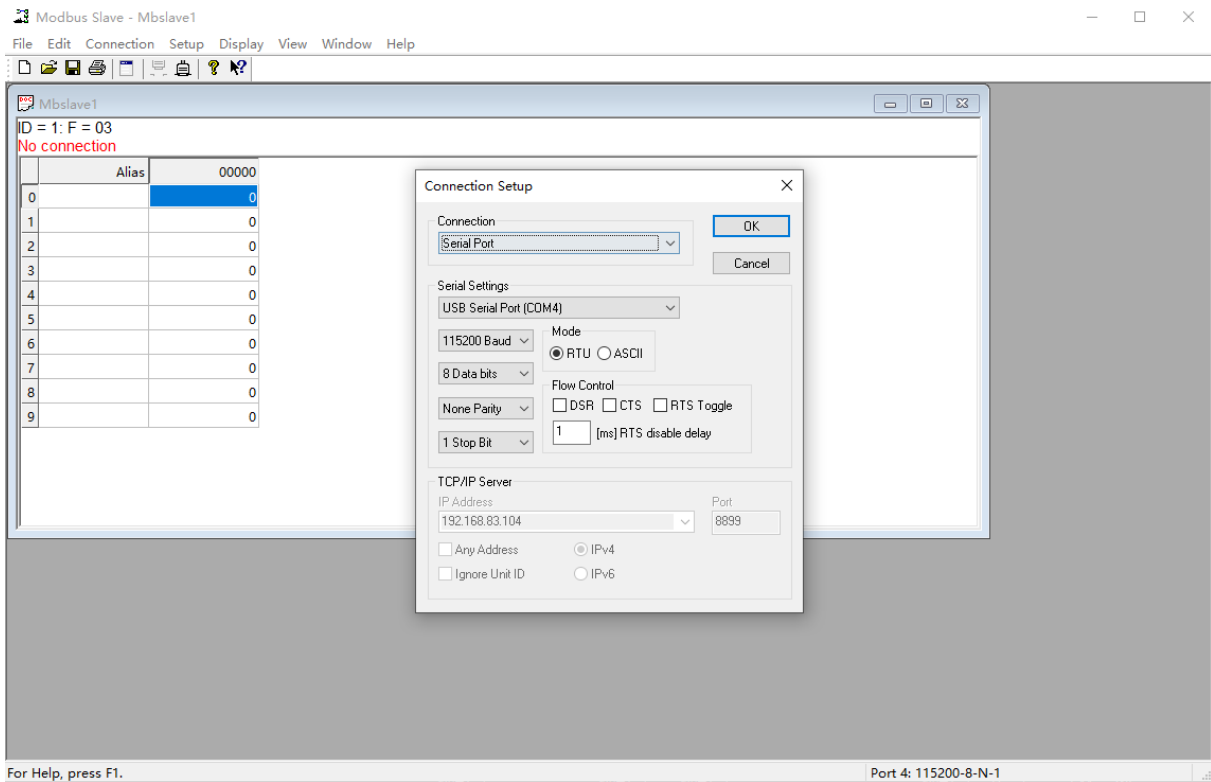
Set server address and reboot(Our MQTT test server: 112.124.43.15, port 1883), it also support HTTP/WebSocket communication.



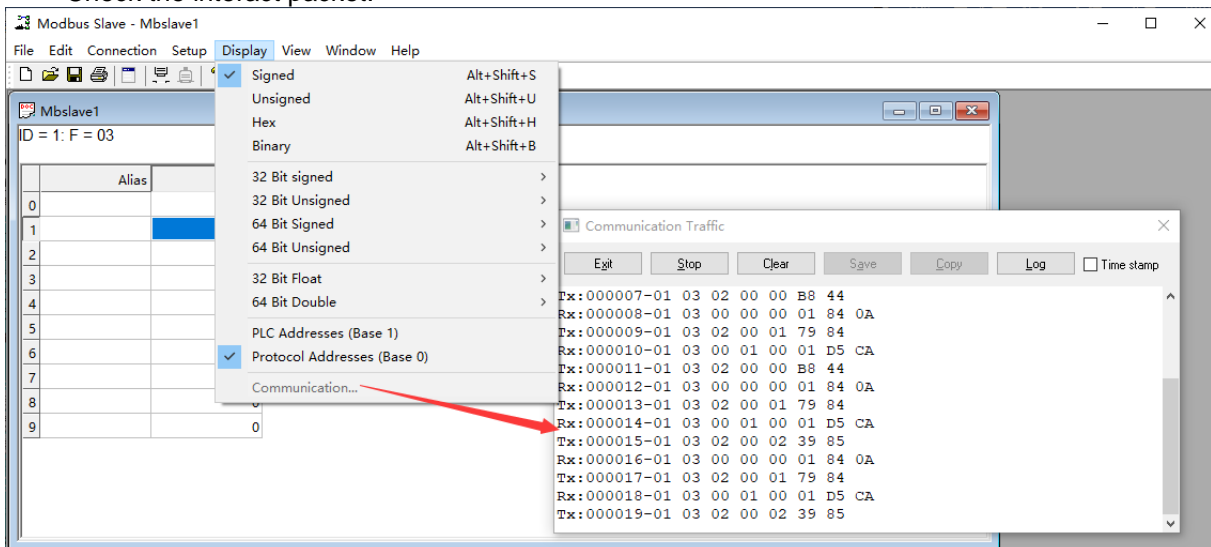
Open UART tools, the device will send out query command as setting, each command interval is 1 second, each command group interval is 10 seconds.



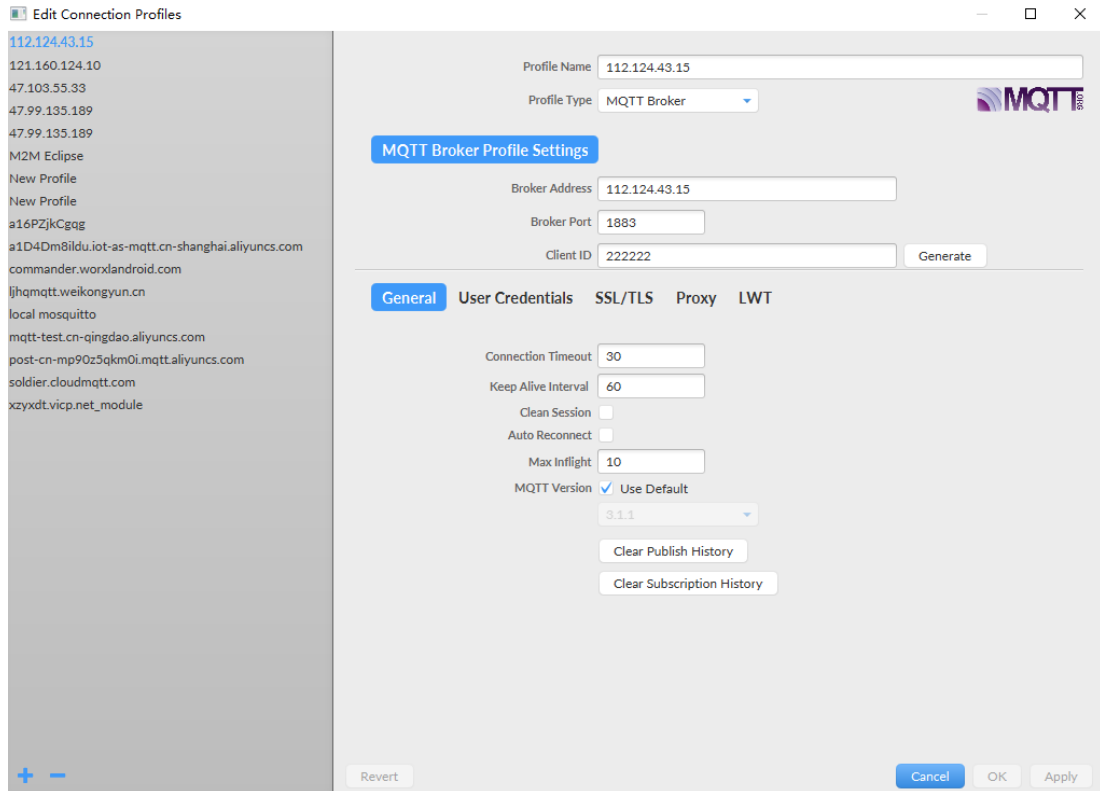
Open Modbus Slave to immitate modbus slave device.



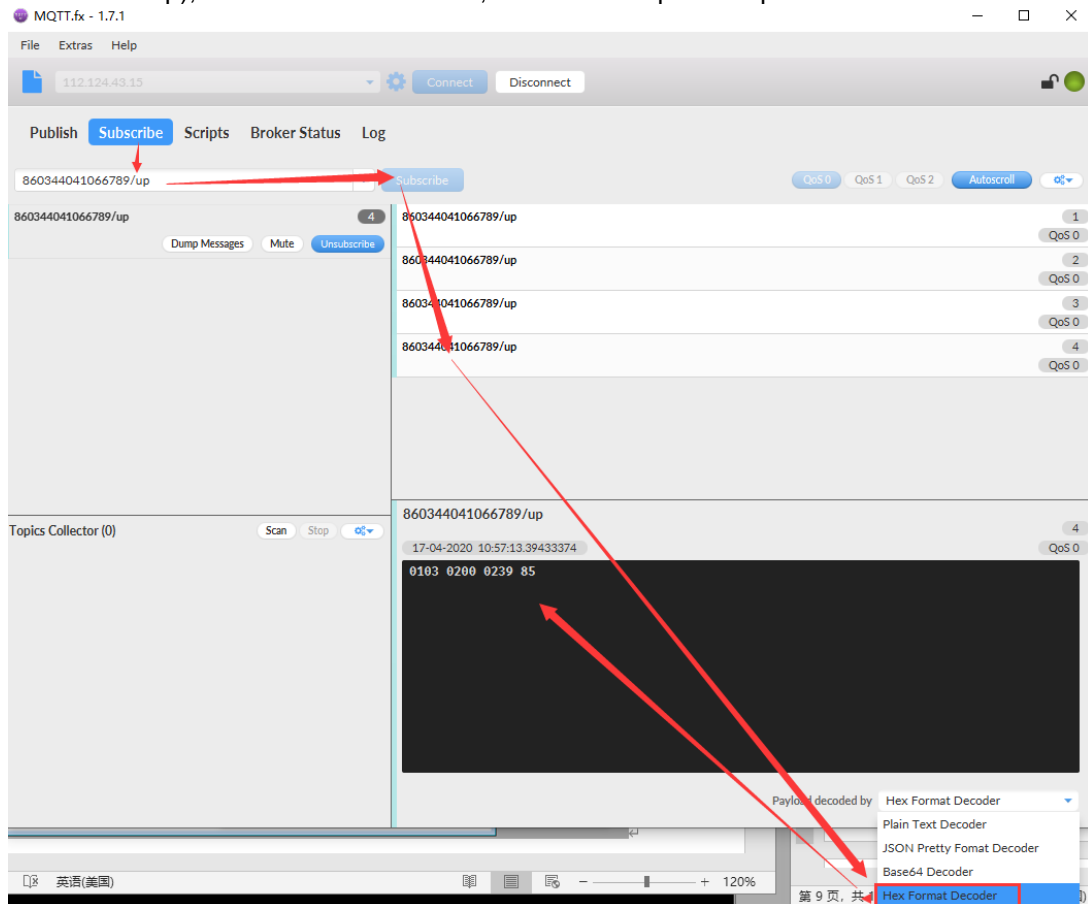
Check the interact packet.



Open MQTT.FX tools, connect to our test MQTT server.



Subscribe the device publish topic(860344041066789/up, device side config use pattern character %IMEI/up), set data as HEX format, then will the uploaded packet



Bound device in IOTService(See details in guide doc). Check device status ,there are Packets count in count.

I.O.T Service


Management (M) Setting (C) Help (H)

Serial Config Config Status VirPath Connected

SN	DevType	MAC Address	HostName	IP	Position	VirPath	Status	SW Ver
7	EG11	300037617507	龚宇伟4	39.144.18.133	China.Pudong		Online	2.0.6c
8	EG11	344049592380	理发店	39.144.4.246	China.Pudong		Online	2.0.6c
9	EG11	344041066789	Gport-EG11	221.178.124.144	China.Pudong		Online	2.0.6 New Ver

Device Status

System



Product ID: EG11
Software Version: 2.0.6
RTC Time: 2020-4-17 11:3:2 Fri
Up Time: 0-Day 0:21:1
Longitude: 121.614395
Latitude: 31.244295

GSM

ModuleSN: 344041066789
ICCID: 89860439101880606970
IMEI: 860344041066789
Connect: Connected(21)
IP Address: 221.178.124.144

SOCKET

SOCKET Name: A
Protocol: MQTT
Status: Connected
Server IP: 112.124.43.15
Recv Bytes: 0 Recv Frames: 0
Send Bytes: 1148 Send Frames: 164
Fail Bytes: 0 Fail Frames: 0

UART

UART No: UART
Config: 115200,8,1,NONE
Recv Bytes: 1148 Recv Frames: 164
Send Bytes: 2010 Send Frames: 251
Fail Bytes: 0 Fail Frames: 0

Reload Restart Edit

Set UploadJson to 1(Upload data as JSON format)

Device Setting

System

Welcome: Gport-EG11
HostName: Gport-EG11
Longitude: 0.0
Latitude: 0.0
IOT Time: 0:0 ~ 23:59

UART

UART No: UART 1
Baudrate: 115200
Data Bits: 8
Stop Bits: 1
Parity: NONE
Flow Control: Disable
UART Protocol: NONE
HeartBeat Time: 0
HeartBeat Serial: IOTWORKSHOP

SOCKET

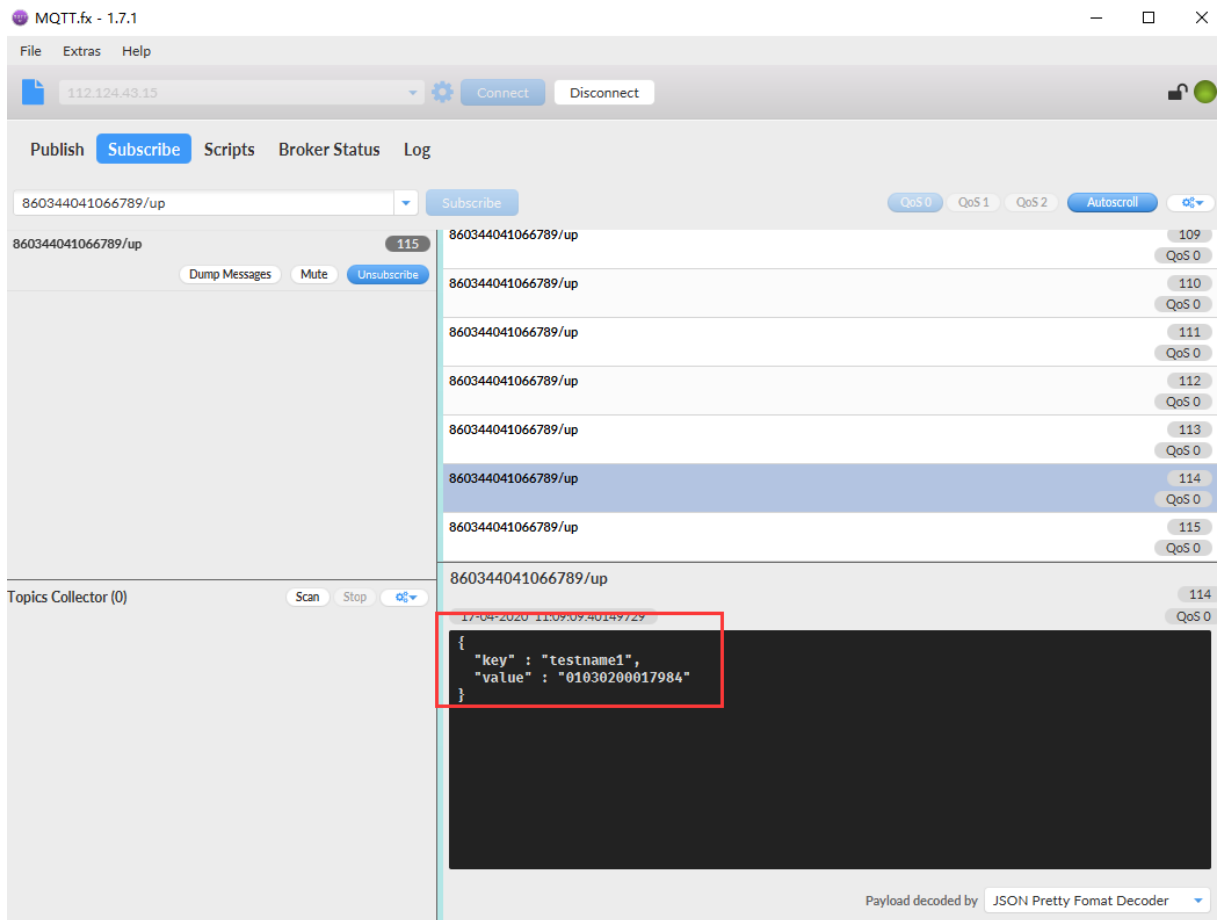
SOCKET Name: A
Protocol: MQTT
Server Addr: 112.124.43.15
Server Port: 1883
Connect Method: Edit Script

Edit Script

HFScriptFunction: NUM 1
QueryInterval(ms): NUM 1000
UploadInterval(s): NUM 10
UploadMethod: NUM 0
UploadJson: NUM 1
JsonName: STRSTR testname1 testname2
QueryCommand1: STRHEX 01 03 00 00 01 84 0A
QueryCommand2: STRHEX 01 03 00 01 00 01 D5 CA

Import Script Export Script Delete Script Read Script Para Add Script Para Confirm Close

Check MQTT.FX received data as following.



Note:

1, This function use HIS script to cofnig device parameters, but actually HIS Script can do more than this such as modify the received UART packet. See more in following link(Except EG11, EG10, HF2111A, PG11, these products does not support this HIS full function as the following link)
<http://www.hi-flying.com/download-center-1/application-notes-1/download-item-his-script>

2.2. Ethernet and Wi-Fi Products Test

Ethernet and Wi-Fi products use IOTService tools to config via network, the detailed products supports this function is as following, this case take HF5111B for example.

Ethernet IOT

FreeRTOS Embedded Network Device
[Eport-E20-PIN] [Eport-E20]
[Eport-E30]
Linux Embedded Network Device
[Eport Pro-EP20-PIN]
[Eport Pro-EP20]
[Eport Pro-EP40]
Ethernet Serial Server
[HF5111S] [HF5111A]
[HF5111B]
Multiple Port Ethernet Serial Server
[HF5122] [HF5142A]
[HF5142B]

Wi-Fi IOT

Wi-Fi Serial Module
[Wport-W20] [Wport-W10]
Wi-Fi Serial Server
[HF2211S] [HF2211]
Multiple Port Wi-Fi Serial Server
[HF2221]
Wifi router (rail)
[HF8104W]

PLC IOT

4G+Ethernet+Wi-Fi+serial
[HF9624]
4G+Ethernet+Wi-Fi
[HF9606]
Ethernet+Wi-Fi
[HF9606W]
Ethernet+Wi-Fi+serial
[HF9610]

4G IOT

4G Serial Server
[HF2411]
4G+WIFI+GPS Serial Server Device
[HF2421G]
4G+WIFI Serial Server
[HF2421]
Rail 4G Router
[HF8102] [HF8104]

Elfin IOT

4G+Ethernet
[Elfin-EG46]
4G_LTE
[Elfin-EG41]
GPRS
[Elfin-EG11&Elfin-EG10]
Wi-Fi
[Elfin-EW10&Elfin-EW11]
Ethernet
[Elfin-EE10&Elfin-EE11]
EMC Protect
[Elfin-EEAX0]
Elfin config
[Elfin-ECA10]

Protoss IOT

4G_LTE
[Protoss-PG41]
GPRS
[Protoss-PG11]
WiFi&Ethernet
[Protoss-PW21]
Wi-Fi
[Protoss-PW11]
Ethernet
[Protoss-PE11]

IO Control

4G IO
[HF6408]
Wi-Fi IO
[HF6208]
Ethernet IO
[HF6508]

2G/NB-IOT

NB-IOT Serial Server
[HF2611]
GPRS Serial Server
[HF2111] [HF2111A]

Upgrade firmware to 1.34.11 version.

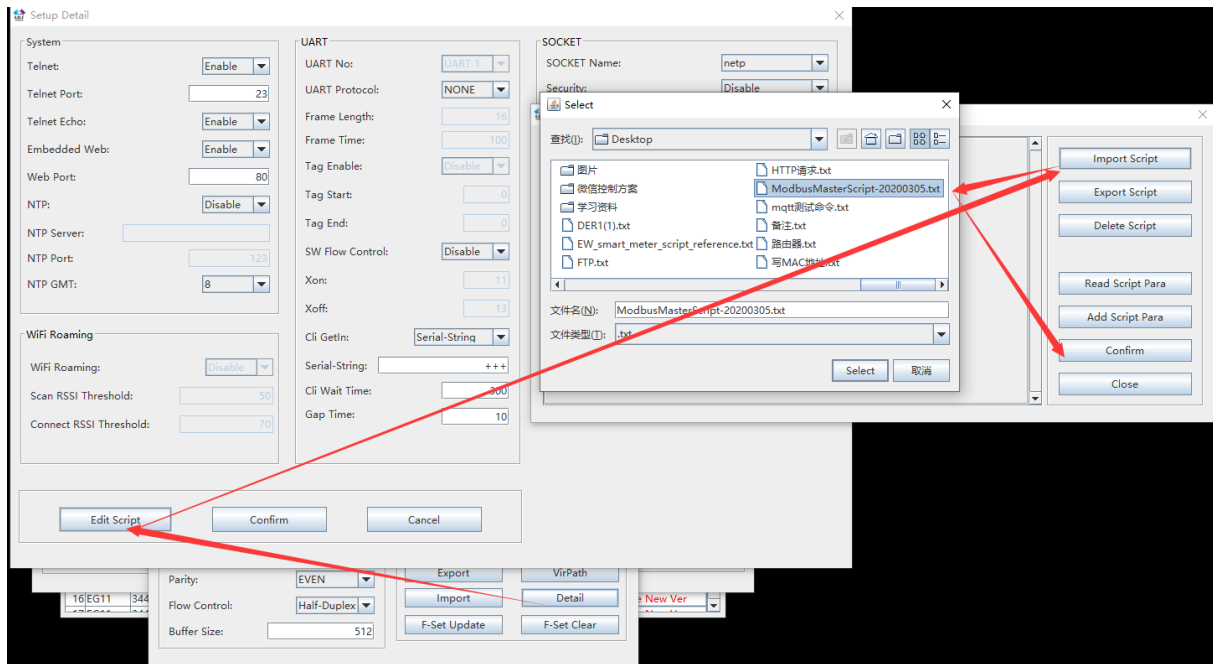
I.O.T Service

Management (M) Setting (C) Help (H)

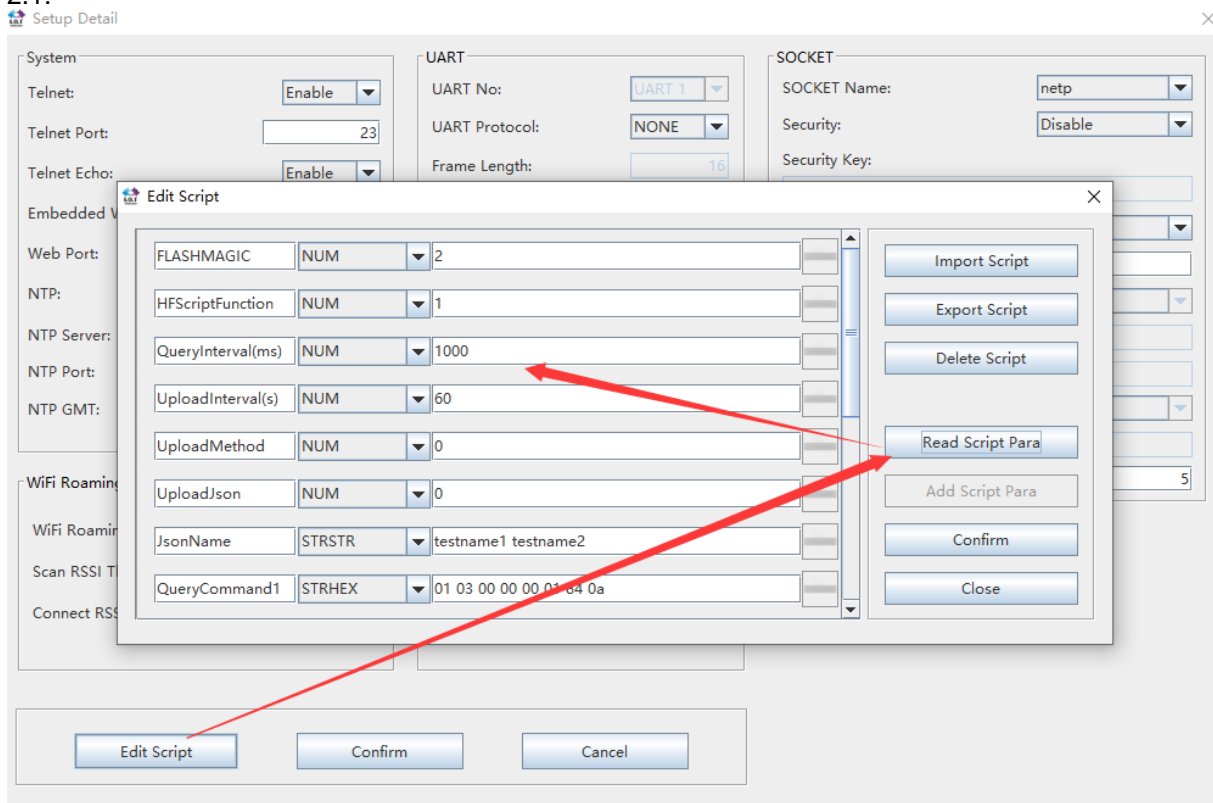
Serial Config Config Status VirPath Connected

SN	DevType	MAC Address	HostName	IP	Position	VirPath	Status	SW Ver
1	HF5111B	F0FE6B111122	HF5111B	192.168.83.105	Local		Online	1.34.11

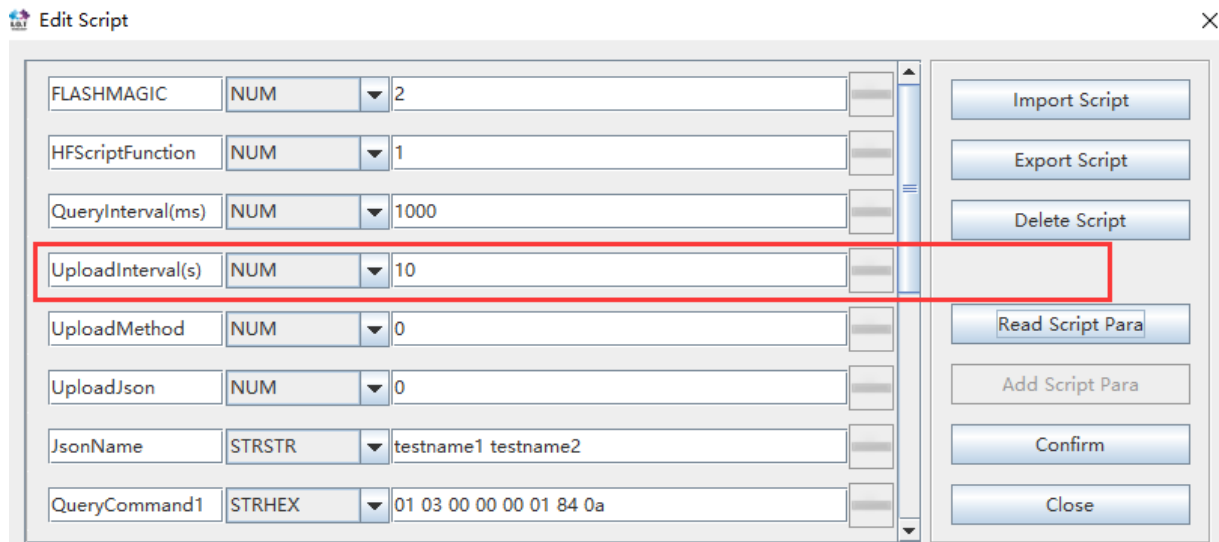
Load config Script file(ModbusMasterScript).



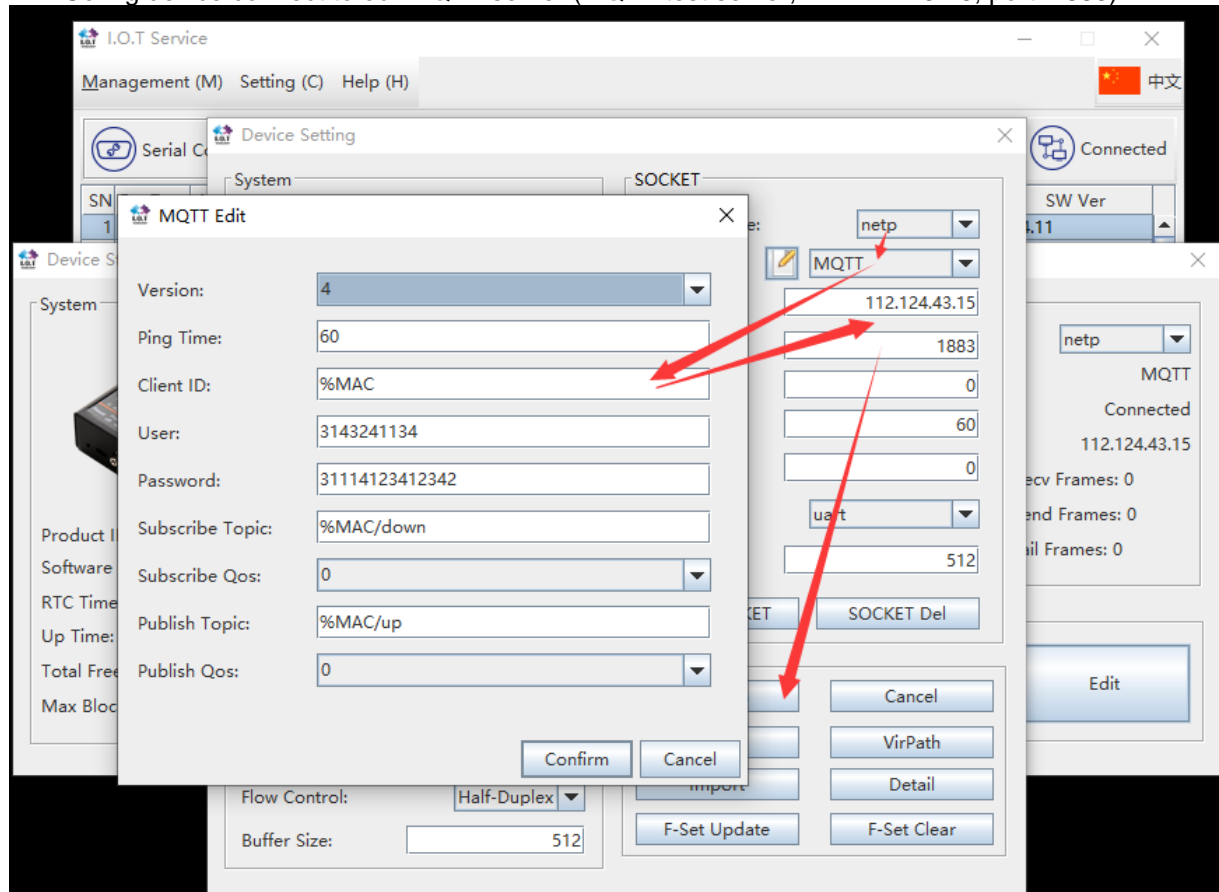
Read Script Para. The detailed parameter function is the same as previous described. in chapter 2.1.




Set UploadInterval to 10 for fast query.



Config device connect to our MQTT server (MQTT test server, 112.124.43.15, port: 1883)



Check device Status that shows it connect to the server.



Product ID: HF5111B
Software Version: 1.34.11
RTC Time: NTP Disabled
Up Time: 0-Day 0:8:12
Total Free Memory: 9548
Max Block Size: 6908

Network

HostName: HF5111B
DHCP: Enable
IP Address: 192.168.83.105
Mask: 255.255.255.0
Gate Way: 192.168.83.1
MAC Address: F0FE6B111122

UART

UART No: UART 1
Config: 115200,8,1,NONE
Recv Bytes: 0 Recv Frames: 0
Send Bytes: 128 Send Frames: 16
Fail Bytes: 0 Fail Frames: 0

SOCKET

SOCKET Name: netp
Protocol: MQTT
Status: Connected

Server IP: 112.124.43.15
Recv Bytes: 0 Recv Frames: 0
Send Bytes: 0 Send Frames: 0
Fail Bytes: 0 Fail Frames: 0

Reload Edit
Restart

Open Modbus Slave to immlute modbus slave device.

Modbus Slave - Mbslave1

File Edit Connection Setup Display View Window Help

Mbslave1

ID = 1: F = 03
No connection

	Alias	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

Connection Setup

Connection: Serial Port

Serial Settings

USB Serial Port (COM4)

Mode: ☒ RTU ☐ ASCII

Flow Control: ☐ DSR ☐ CTS ☐ RTS Toggle

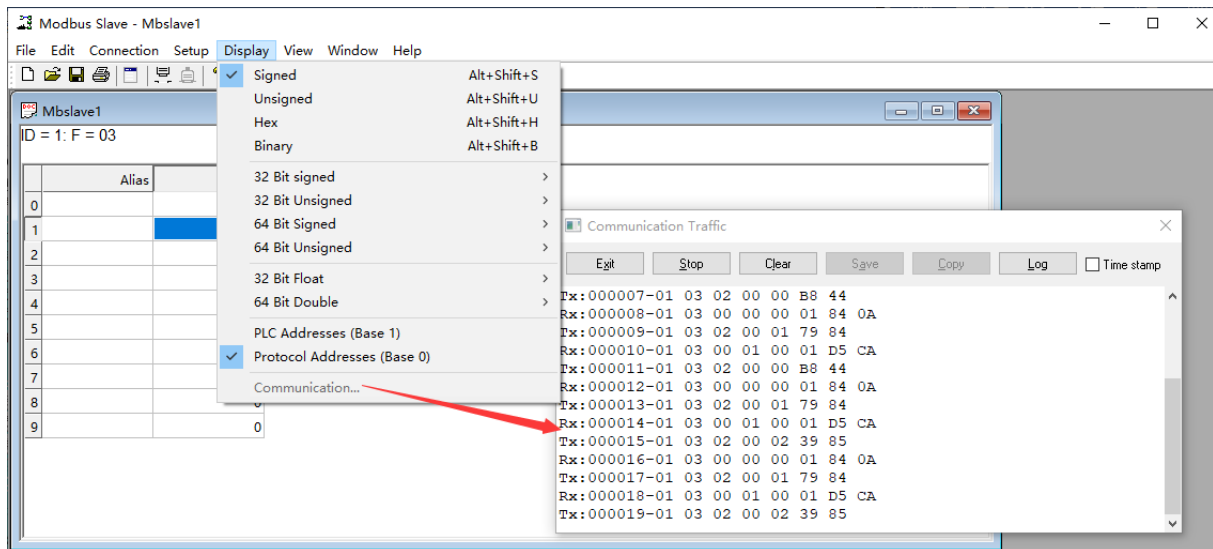
1 Stop Bit

TCP/IP Server

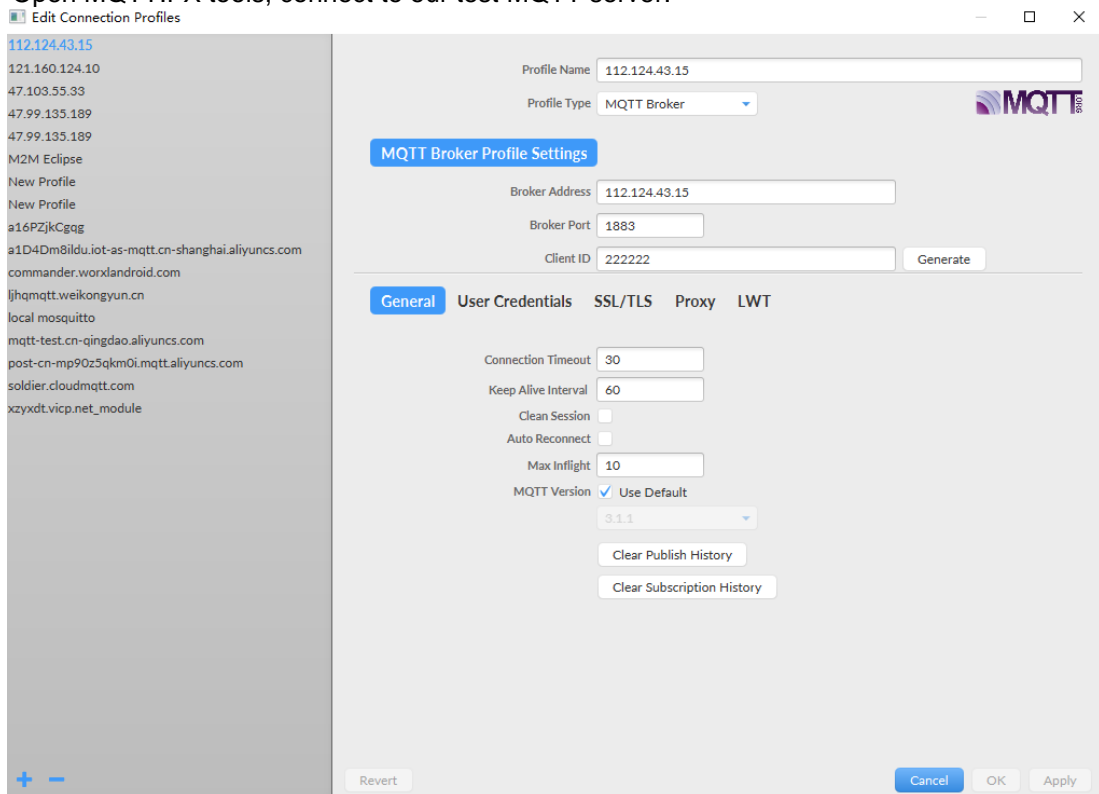
IP Address: 192.168.83.104 Port: 8899

☐ Any Address ☒ IPv4
☐ Ignore Unit ID ☐ IPv6

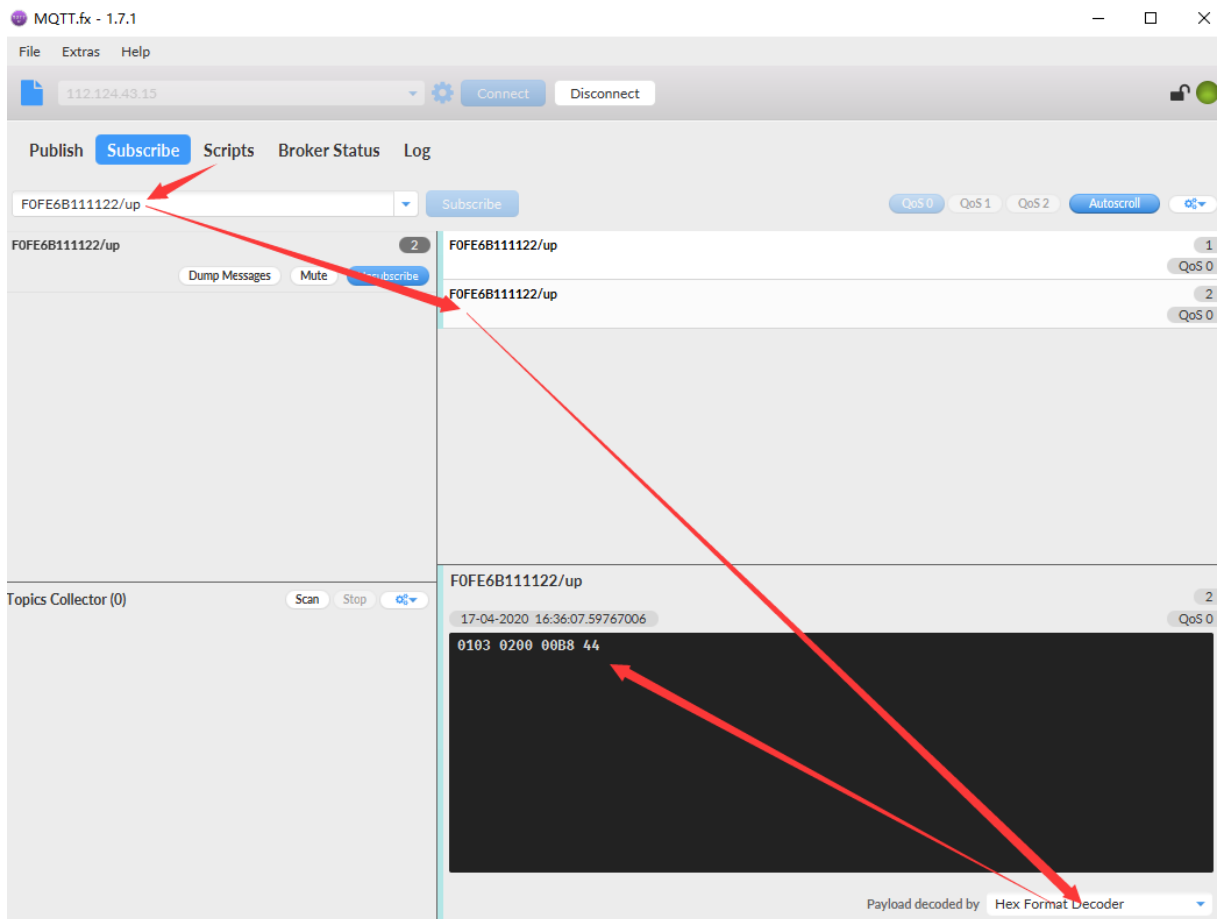
Check the interact packet.



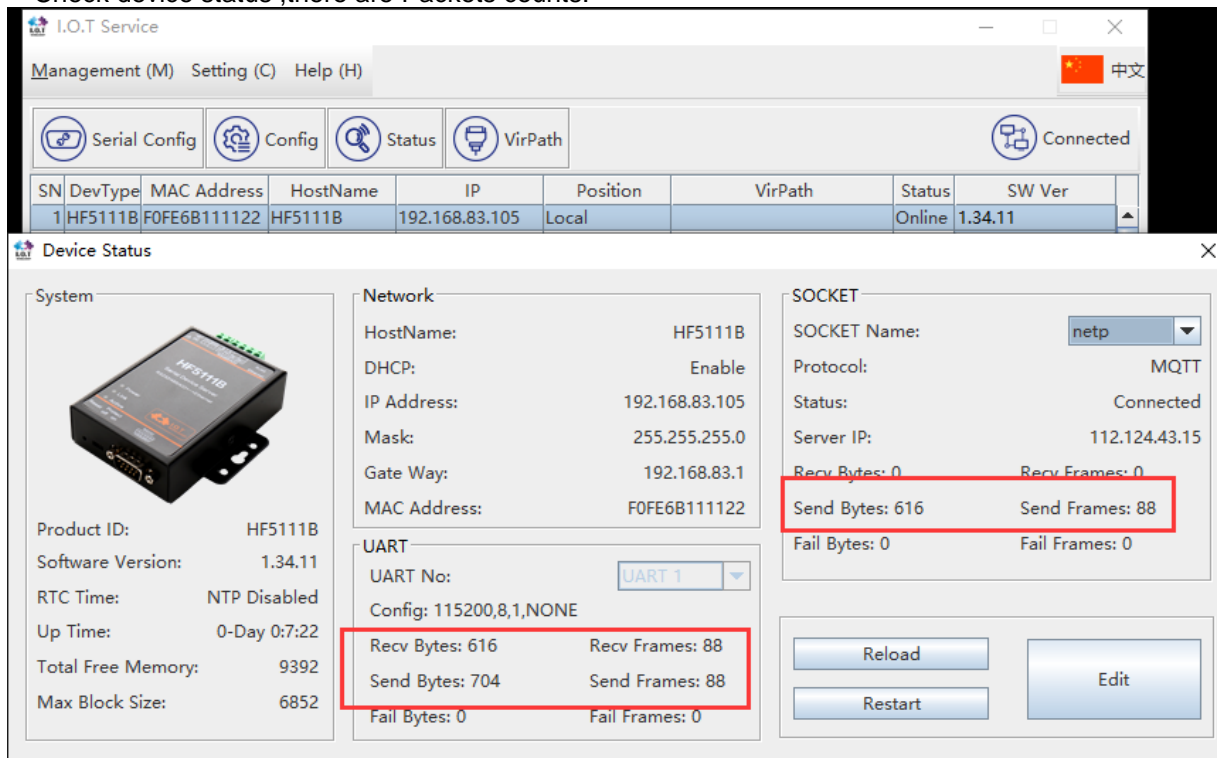
Open MQTT.FX tools, connect to our test MQTT server.



Subscribe the device publish topic(860344041066789/up, device side config use pattern character %IMEI/up), set data as HEX format, then will the uploaded packet



Check device status ,there are Packets counts.



Set UploadJson to 1(Upload data as JSON format)

Edit Script

FLASHMAGIC	NUM	2
HFScriptFunction	NUM	1
QueryInterval(ms)	NUM	1000
UploadInterval(s)	NUM	10
UploadMethod	NUM	0
UploadJson	NUM	1
JsonName	STRSTR	testname1 testname2
QueryCommand1	STRHEX	01 03 00 00 00 01 84 0a

Buttons: Import Script, Export Script, Delete Script, Read Script Para, Add Script Para, Confirm, Close

Check MQTT.FX received data as following.

MQTT.fx - 1.7.1

File Extras Help

112.124.43.15 Connect Disconnect

Publish Subscribe Scripts Broker Status Log

860344041066789/up Subscribe QoS 0 QoS 1 QoS 2 Autoscroll

Topic	Message ID	QoS	Payload
860344041066789/up	115	QoS 0	{ "key" : "testname1", "value" : "01030200017984" }

Topics Collector (0) Scan Stop

Payload decoded by JSON Pretty Format Decoder

Note:

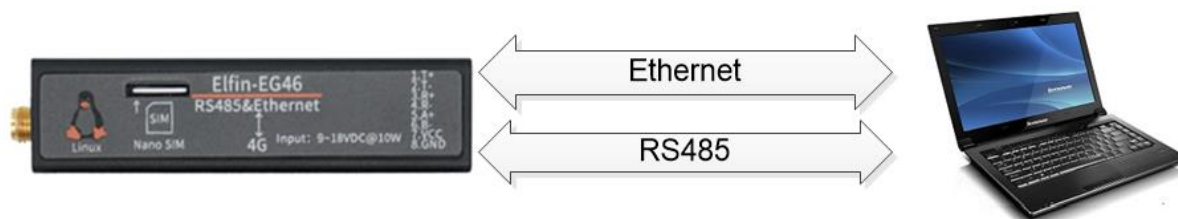
1, This function use HIS script to cofnig device parameters, but actually HIS Script can do more than this such as modify the received UART packet. See more in following link)

<http://www.hi-flying.com/download-center-1/application-notes-1/download-item-his-script>

3. COMPLETE MODBUS MASTER TEST

3.1. MODBUS MASTER POLLING FUNCTION

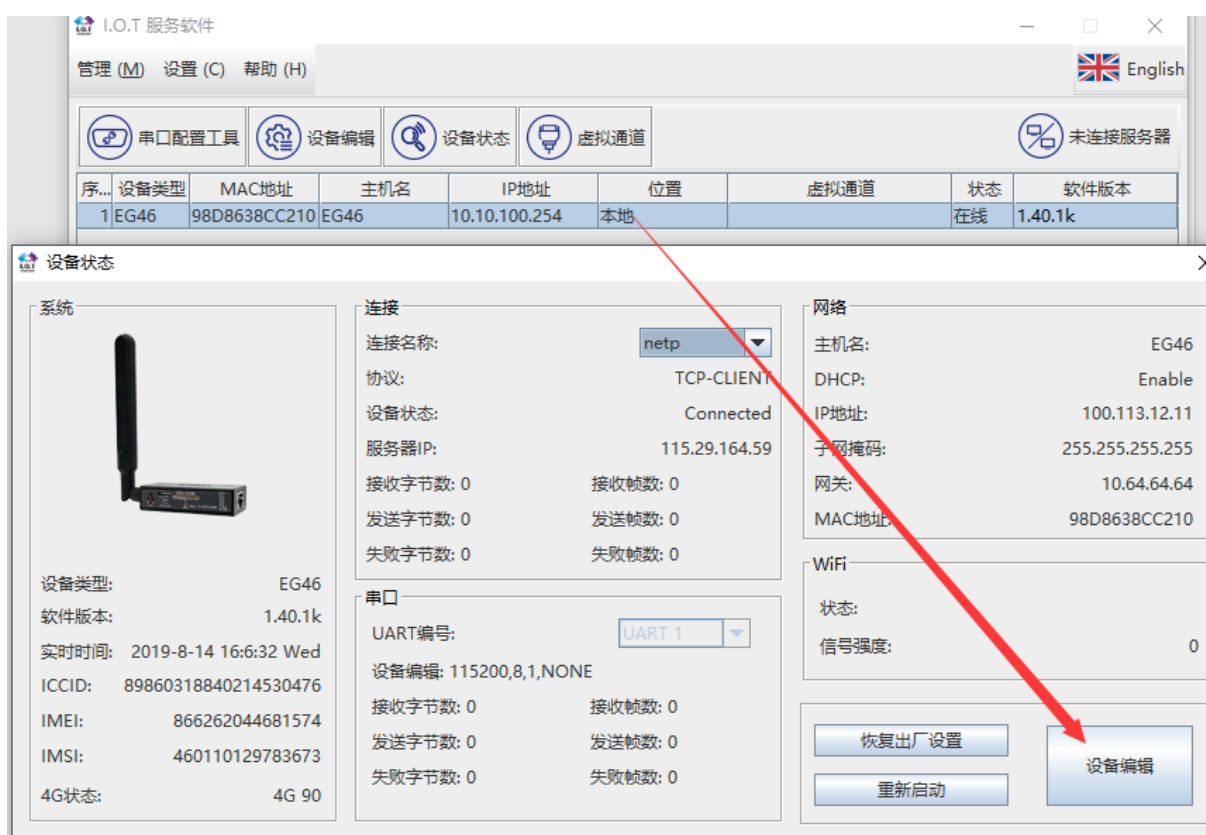
PC network cable and serial cable connection EG46



Ethernet LAN IP: 10.10.100.254

IP:10.10.100.173

Open the IOTService Tool and click Edit.



Modify the serial port protocol to device (Modbus master function, the serial port cannot use the cli command in this configuration state).

高级设置

系统

Telnet:

Telnet端口:

Telnet回显:

内嵌网页:

内嵌网页端口:

NTP:

NTP服务器:

NTP端口:

NTP时区:

串口

UART编号:

UART协议:

帧长:

成帧时间:

成帧标记:

开始标记:

结束标记:

软件流控:

Xon:

Xoff:

CLI进入方式:

Serial-String:

CLI超时时间:

Gap Time:

连接

连接名称:

数据加密:

加密密码:

连接模式:

断开连接字符串:

心跳使能:

心跳时间:

心跳包:

注册包模式:

注册包:

最大连接数Max Client ...

Right click on the device information and click on the Modbus settings page.

I.O.T 服务软件

管理 (M) 设置 (C) 帮助 (H)

English

串口配置工具 设备编辑 设备状态 虚拟通道

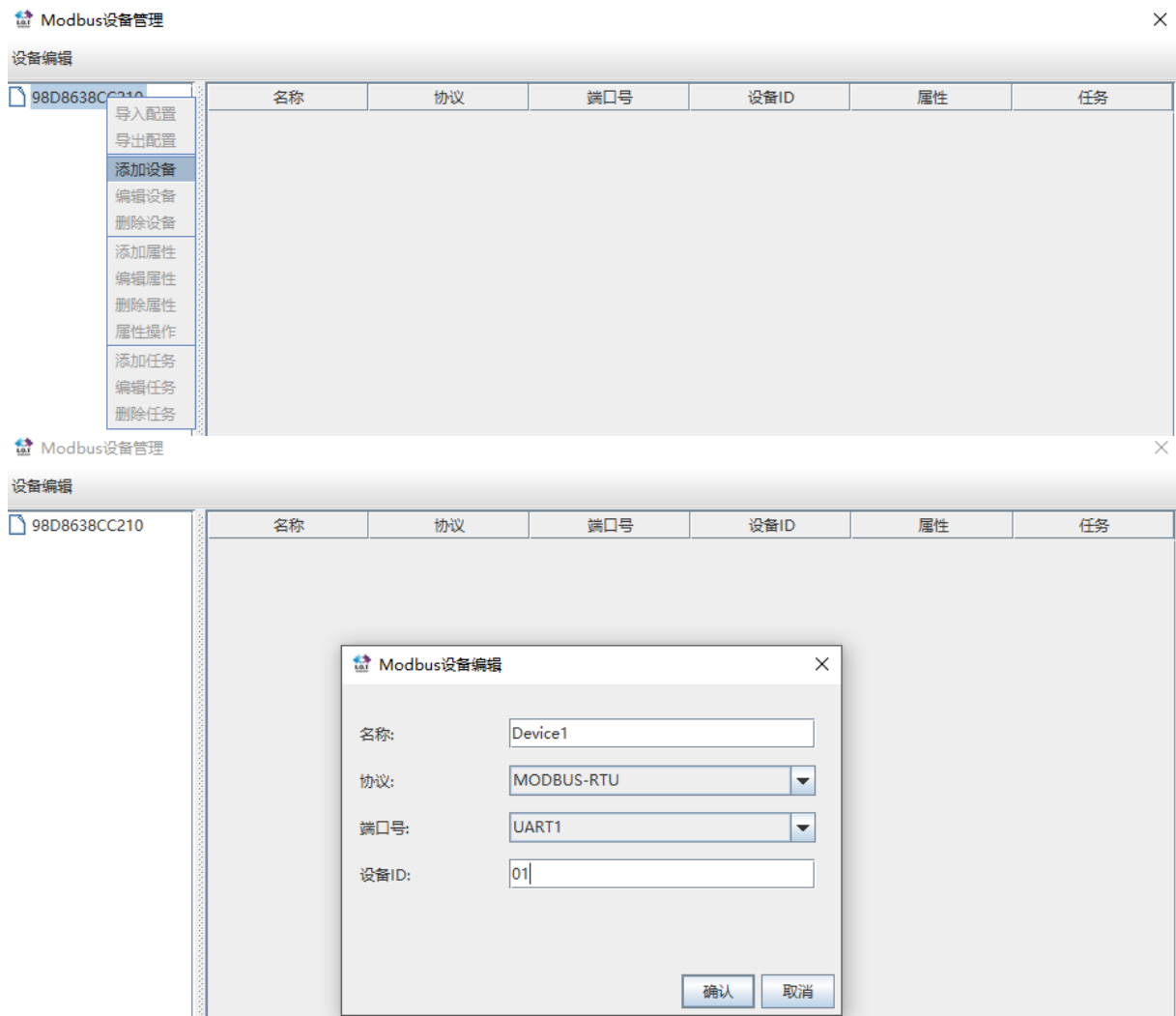
未连接服务器

序...	设备类型	MAC地址	主机名	IP地址	位置	虚拟通道	状态	软件版本
1	EG46	98D8638CC210	EG46	10.10.100.254			在线	1.40.1k

复制MAC到剪切板
设备列表过滤
刷新列表
删除已选设备
升级选择设备的程序
升级所有设备的程序
升级选择设备的网页
本地设备保存出厂参数
应用程序 ▶ Modbus设置

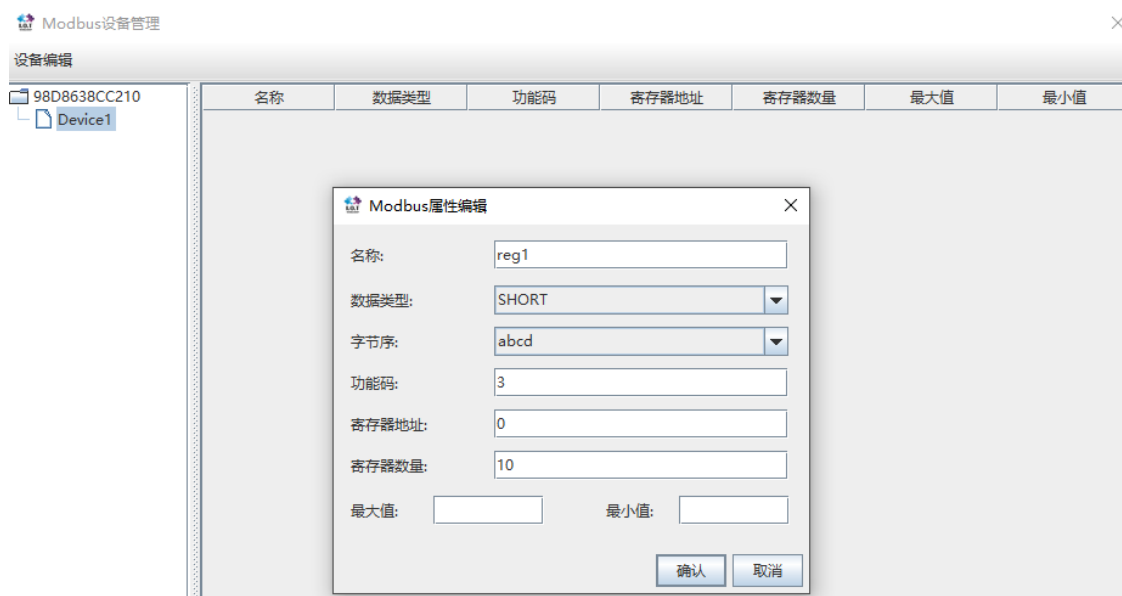
Right click on EG46 and select Add Device.

- Name: Modbus slave name
- Protocol: Modbus slave device protocol type
- Port number: Serial port number (EG46 only 1 serial port, fill in UART1)
- Device ID: Modbus slave address



Right-click the created Device device and select Add Attribute.

- Name: The name used when data is reported. When polling multiple registers, the data with no name is automatically reported according to the regXX name.
- Data type: short, bool or float, affecting data reporting.
 - bool: The report is false or true.
 - short: The data is short, and the length of 2 bytes is used as a register value.
 - float: The data is in floating point number and the length of 4 bytes is used as a register value.
- Endian: The byte order of the data, whether the high order is in the front or the low order is in the front, and so on.
 - abcd: big endian mode
 - badc: big endian mode and swap byte order.
 - dcba: little endian mode
 - cdab: little endian mode and swap byte order.
- Function code: The function code of modbus polling.
- Register address: modbus polling register start address
- Number of registers: The number of modbus polling registers.
- Maximum: The maximum data allowed under the float data type.
- Minimum value: The minimum value of data allowed under the float data type.

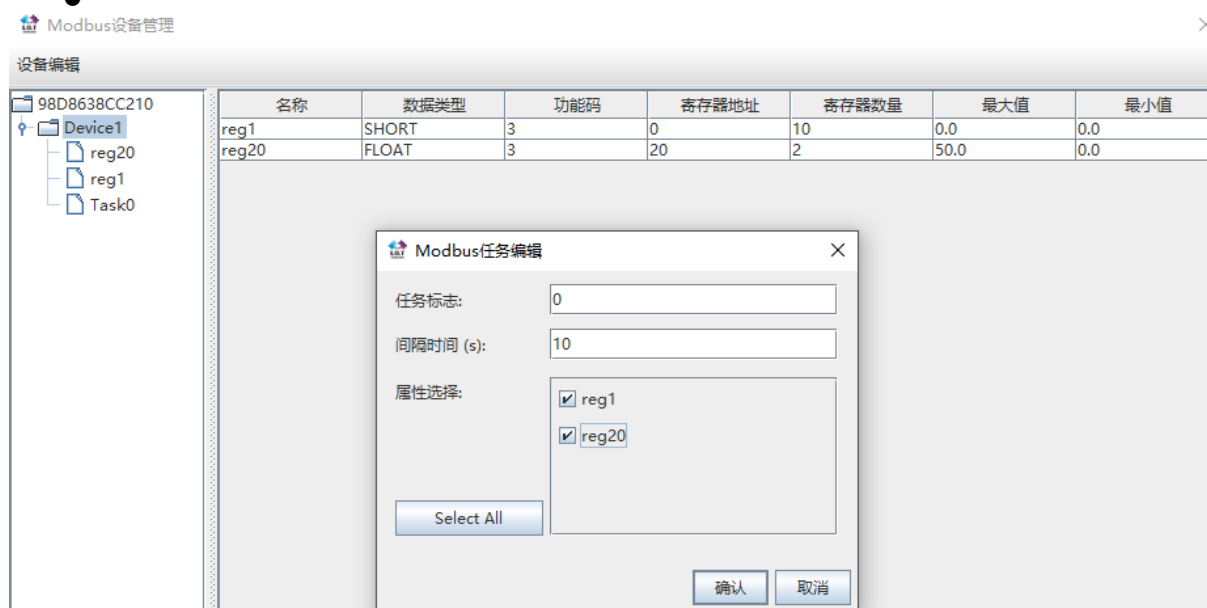


Two query instructions are created as follows, reg20 is of type float.



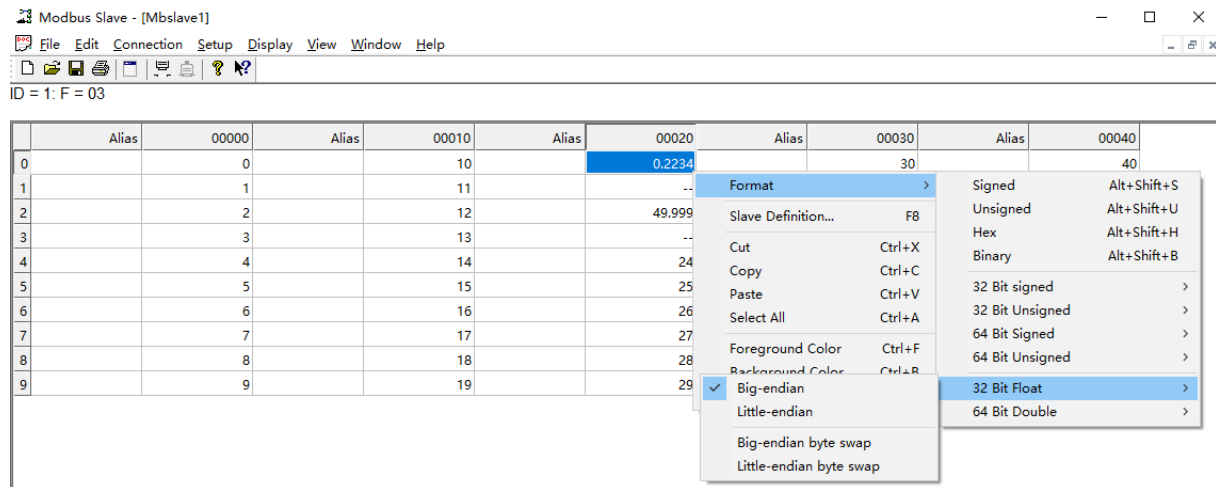
Create a task, set the collection time and collection range. After the task is created, you need to restart the device to take effect.

- Task tag: Task id name
- Interval: A complete polling cycle.
- Attribute selection: The name of the register participating in the polling.

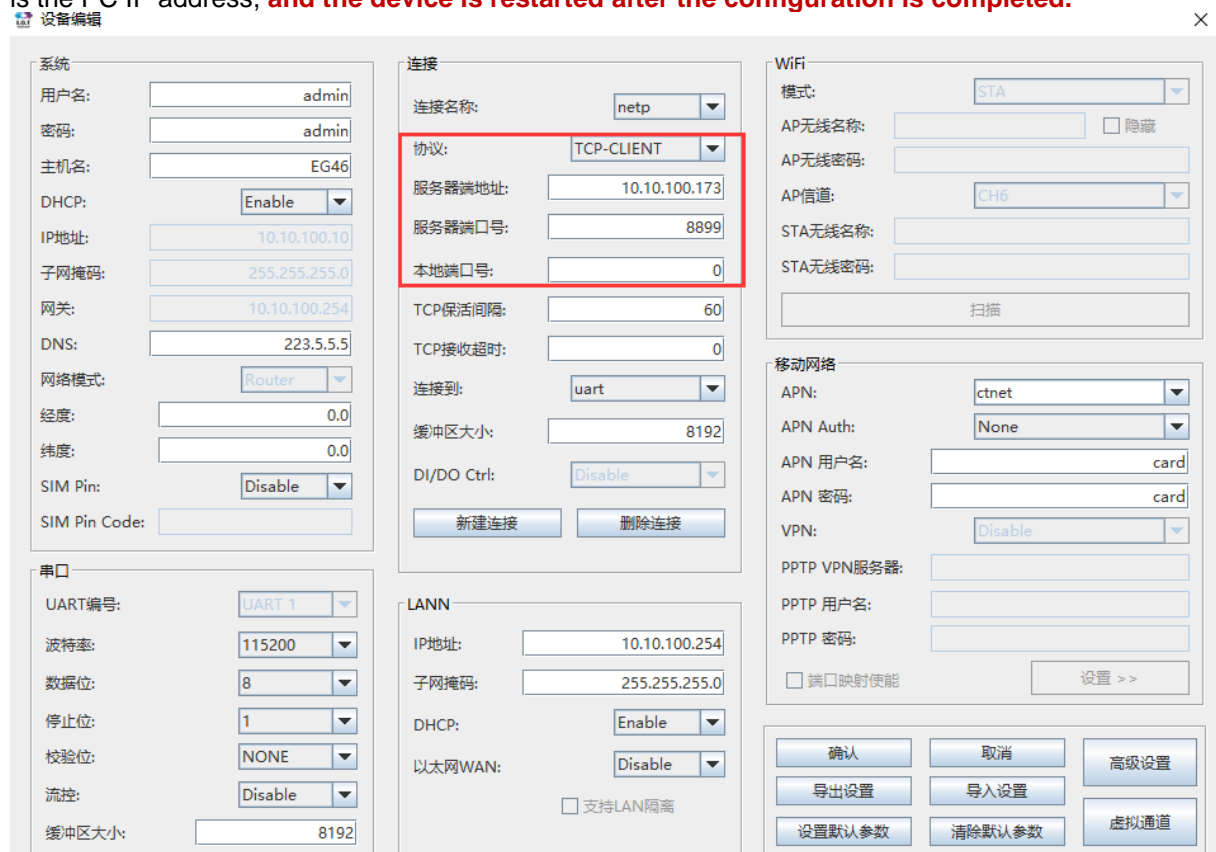


The Modbus slave software is opened on the PC side, and the Modbus device is simulated. As shown in the figure below, the address is set to 1, the function code is 3, the register value is set as

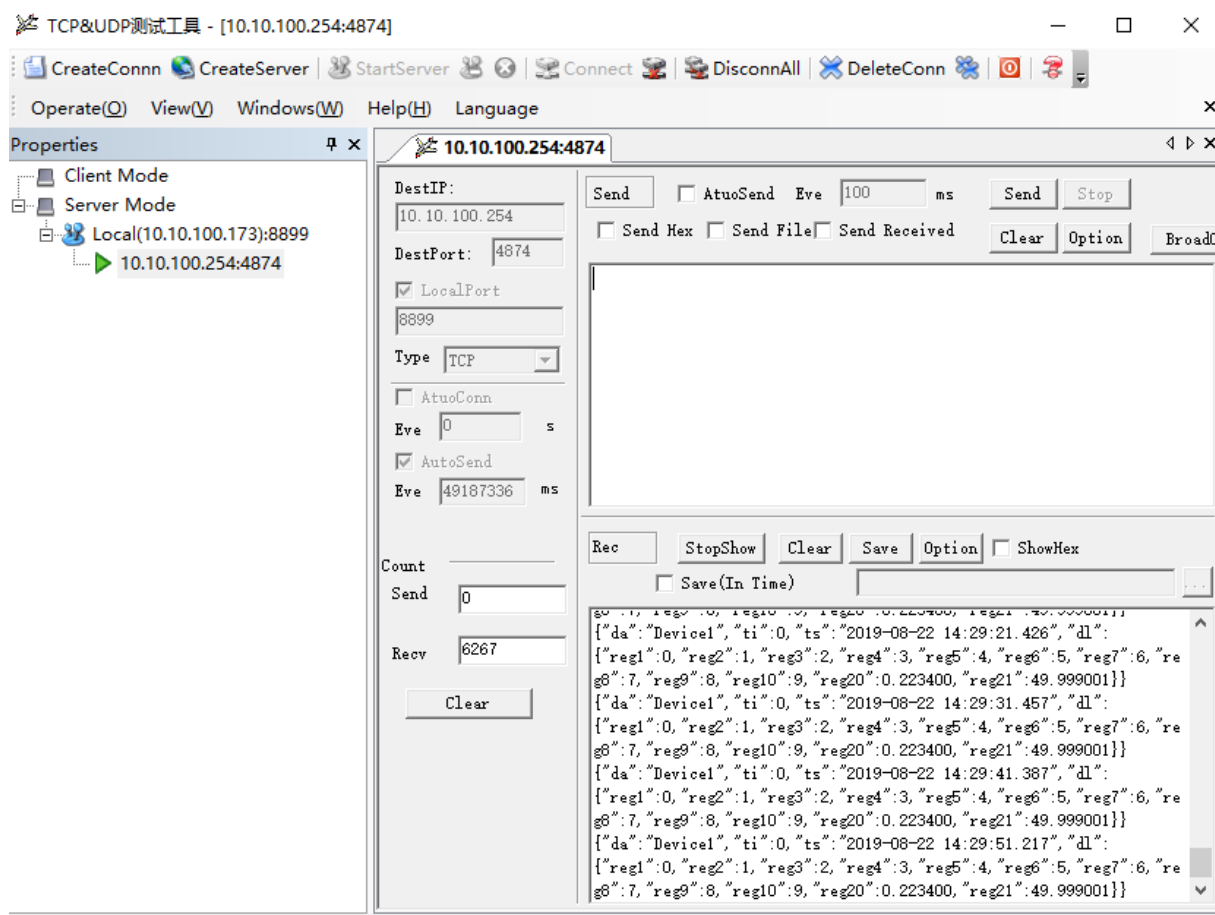
shown below, and the data of the address 20 and 22 is configured as the big end floating point number.



After the above configuration, the serial port is set up. After that, you can set the data report. You can use TCP, HTTP, MQTT, etc. The following example uses TCP Client to connect to the PC to view the data reported after polling. The aircraft software establishes TCP Server, port 8899, 10.10.100.173 is the PC IP address, **and the device is restarted after the configuration is completed.**



From the PC side, you can see the message sent after the device polls.



The interactive process is described as follows:

According to the configuration of the serial port in the above example, the product automatically initiates polling data every 10 seconds, and a single poll initiates two instructions.

Instruction 1: modbus address 1, function code 3, register start address 0, length 10.

Device Polling Command: 01 03 00 00 00 0A C5 CD

Modbus Slave reply: 01 03 14 00 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 CD 51

Instruction 2: modbus address 1, function code 3, register start address 20, length 4.

Device polling command: 01 03 00 14 00 04 04 0D

Modbus Slave reply: 01 03 08 3E 64 C2 F8 42 47 FE FA E7 F1

For the received data, the data is parsed according to the floating point number or the short type in the configuration, and the configuration information is reported and reported.

```
{
  "da": "Device1",           //设备名称
  "ti": 0,                   //任务标记
  "ts": "2019-08-22 14:14:05.409", //时间戳
  "dl": {
    "reg1": 0,               //寄存器名称, short 类型。
    "reg2": 1,
    "reg3": 2,
    "reg4": 3,
    "reg5": 4,
    "reg6": 5,
    "reg7": 6,
```

```
"reg8": 7,  
"reg9": 8,  
"reg10": 9,  
"reg20": 0.223400,           //寄存器名称，浮点数类型。  
"reg21": 49.999001  
}  
}
```

Remark:

- 1、Support for creating multiple modbus slave devices.
- 2、Support several data points.
- 3、Support batch import and export configuration information, directly import parameters, and complete parameter configuration with one click.

