# RS485 TO POE ETH (B) MQTT And JSON User Manual

From Waveshare Wiki Jump to: navigation, search



(/wiki/File:RS485\_TO\_POE\_ETH01.jpg)

#### **Overview**

MQTT and JSON can be used alone or together. JSON supports Modbusconvert RTU format to JSON format.

Main features:

1. Use the MQTT-based protocol to establish a connection with the server, and use the form of subscription to publish data communication.

2. Support independent design and automatic collection of Modbus RTU registers.

3. Support the conversion of specific Modbus register content into JSON format and send it regularly and actively.

- 4. Support adding device ID, time, and any string in JSON format.
- 5. Support embeddings in JSON format.
- 6. Support NTP protocol, and get the time automatically.
- 7. Support unsigned data and signed data, support decimal point representation, and

support 4-byte length data.

8. All configurations can be completed in interface configuration, and the user's independent configuration does not need to be customized.

# **JSON Examples**

#### MODBUS RTU To JSON

- Modbus RTU to JSON can realize an automatic collection of Modbus RTU tables, and is automatically uploaded to the cloud server following the JSON format.
- Here we explain this usage with a specific case.

#### MODBUS Table

 Suppose there is a Modbus table with a function code of 3 and an address of 1. Its register addresses and parameter names are as follows. Where the byte length is 4, it means that 2 registers need to be read continuously.

Resister Address	Parameter	Byte Length	Note
0	Current total active power	4	Unsigned, keep 2 decimal places
97	Phase A voltage	2	
98	Phase B voltage	2	Unsigned, keep 1 decimal place
99	Phase C voltage	2	
100	Phase A current	2	
101	Phase B current	2	Unsigned keep 2 designed place
102	Phase C current	2	Unsigned, keep 2 decimal place
119	Frequency	2	
356	Phase A active power	4	
358	Pahse B active power	4	Unimed here 2 desired also
360	Phase C active power	4	Unsignea, keep 3 decimal place
362	Total active power	4	

The so-called signed means that the highest bit of 2 bytes or 4 bytes is the sign bit, for example, 0xFFFF will be recognized as -1. Keeping 2 decimal places means that after the data is converted as an integer, the decimal point moves from the rightmost to the left 2 digits.

#### **Device Config**

- We configure the device as a client.
- Use SSCOM to monitor a TCP server on port 1883 of the local computer.

			ita debugger	,Authors (mbh,2	010030	eqq.co		n Deservation of the	-		~
PORT	COM_Settings	Display	Send_Data	Multi_Strings	Tools	Help	联系作者	大虾论坛			_
learB	ata OpenFile				5	andFila	Stop [C]	earSead OnIcol	✓ English Series	vo[onfire	FIT
learD	ata OpenFile			1	5	endFile	Stop Cla	earSea	€ English Sa	veConfig	EIT
learD	ata OpenFile	502 31	► HEXS	thow SaveData	S C Rec	endFile eivedTol	Stop Cl	earSend[□ OnTop] ndHEX □ SendEvery ButasTo]== □ very	✓ English <u>Sav</u> :[50 ms/T	veConfig	EXT . CrLf
learD onNun mot 19	ata OpenFile ICFServer 2168.1.200 2168.1.206	502 <u>11</u>	HEXS	bow <u>SaveData</u> Time and Pack	S F Rec e OverTi	endFile eivedTol n# 20	Stop Cl. File Sec ms No 1	earSend「OnTop] ndEX「SendEvery BytesTo末尾 ▼Vez	✓ English See :50 ms/T iEyNone	veConfig im Add	EXT . CrLf
learD onNun mot 19 ocal 19	ata OpenFile TCFServer 12 168 1.200 12 168 1.206	502 113 1883 Dis	▼ F HEXS	bow <u>SaveData</u> Time and Pack	S Rec e OverTi	endFile eivedTol ne 20	Stop Cl. File Sec ms No 1	earSend OnTop ndHEX 「SendEvery BytesTo 末尾 ▼Ver	✓ English Sar 50 ms/T i£yNone	veConfig im Add	EXT
learD onNun mot 19 了更好 您注册	ata OpenFile TCFServer 21681.200 21681.206 业发展SSCON分件 素立创时结尾客户	502 <u>16</u> 1883 <u>pis</u>	Thexs ten Show	how <u>SaveData</u> Time and Pack	S Rec e OverTi	endFile eivedTol ne 20	Stop Cla File Se ms Bo 1	earSend OnTop ndHEX 「SendEvery BytesTo 末尾 ▼Ver	⊽ English Se :50 ms/T i£yNone	veConfig im/ Add	EXT
learD mot 19 real 19 了更好 你主册 升级到	ata OpenFile TCFServer 2168.1.200 2168.1.206 业发展SSCON软件 嘉立创时结尾客户 财5.13.1】★合言	502 <u>1i</u> 1883 Dis 5 <b>SE</b> 高性价比4	▼ HEXS tten Show	bow <u>SaveData</u> Time and Pack € ★RT-Thread⊄	S F Rec • OverTi	endFile eivedTol ne 20	Stop Clo File See ms No 1	earSend OnTop ndHEX 「SendEvery BytesTo 末尾 ▼Ver 新一代WiFi芯片兼	✔ English <u>Sa</u> 50 ms/T ifyNone 音8266支持和	veConfig in Add	EXT CrLf
LearD mbun not 19 cal 19 了里好 影注册 升级型 ww.da:	ata OpenFile TCPServer 12 168 1 200 2168 1 200 2168 1 200 地发展SSCON软件 嘉立创时结尾客户 W5. 13. 1 】 ★合言 xia.com S:0	502 1883 Dis 5 5 5 5 5 8 5 8 7 6 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	▼ F HEXS conf F Show conf F Show	how <u>SaveData</u> Time and Pack K <b>X</b> RT-Thread CPserver [Liste	S F Rece e OverTi 和国人的开	en.dFile eivedTol ne 20 F源免费 ellP=0.0	Stop Cl File Sec ms No 1	earSen OnTop ndHEX 「SendEvery BytesTo 末尾 Vet 就一代WiFi芯片兼 本机所有IP都在侦明	✓ English <u>Sa</u> 50 ms/T i £yNone 容8266支持87	veConfig im厂 Add 一Thread 显示本地	EXT CrLf #8EM

• Configure the device via Vircom:

Device Info	12 000	Network		Advanced Setting	s	1	1
Virtual Serial	Not Use 💌	IP Mode	Static	DNS Server IP	8.8.4	. 4	Auto Search
Dev Type		IP Address	192 . 168 . 1 . 200	Dest. Mode	Dynamic	-	
Dev Name	WSDEV0001	Port	0	Transfer Protocol	None	•	Add Manually
Dev ID	28666CF390A2	[]  Work Mode	TCP Client	Keep Alive Time	60	(s)	
Firmware Ver	V1.452	Net Mask	255 . 255 . 255 . 0	Reconnet Time	12	(s)	Search Serial
E		Gateway	192 . 168 . 1 . 1	Http Port	80		
-Function of the	a device	Dest. IP/Dom	nain 192.168.1.206 Local	IP UDP Group IP	230 _ 90 _ 76	. 1	P2P Device
DMS Sector	ward -	Dest. Port	1883	Register Pkt:	l I	ASCII	
T DEAL OCK	UTA	Serial	- 1941 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 1942 - 194	Restart for no d	tata every 300	Sec.	Edit Device
Modbus TC	P To RTU	Baud Rate	460800 -	Enable send pa	arameter every 6	Min.	Search List
F Serial Com	mned	Data Bits	8	More Adv	raced Settings		
C DHCP Sum	nort	Parity	None	-			Back
Storage Ex	tend	Stop Bits	1	Framing Rule Max Frame Length	1300	(Byte)	
R MUNHTOP O	Connection	Flow Control	None	Max Interval(Small	er will better) 3	(Ms)	

(/wiki/File:RS485\_TO\_POE\_ETH04.jpg)

 Click Modify configuration to connect the device to the SocketDlgTest tool. Enter device editing again dialog box. Click the "Firmware and Configuration" button.

Weepage directly in local PC:         E:场員VRS485 TO ETH (POE)WQTTWQTTHTTPD         Special configs:       Clear all         ZLMB config       WQTT config         ISON config       Reg packet	•
Special configs: ZLMB config MQTT config ISON config Reg packet Code file download mode Select code file:	<b>•</b>
Special configs: Clear all ZLMB config MQTT confic ISON confic Reg packet C Code file download mode Select code file:	
ZLMB config     WQTT config     ISON config     Reg packet       Code file download mode     Select code file:	
Code file download mode Select code file:	
E1项目VRS232 TO ETHVRS232 TO POE ETH (B)固件WVSDEV0001.bin	*
Download through the network     C Download through serial port	
Device IP address or domain: 192.168.1.200 Serial port: COM1	
Download port (Don't modify): 1092 Baundrate: 115200	
Device modual/type: 2003    DevID: 28666CF390A2 Bind ID	
Flash size: 256 V KB	
Please close the opened webpage of the modual in the browser, before start download.	
Download	

(/wiki/File:RS485\_TO\_POE\_ETH05.jpg)

- First click "Web Directory Download" to enter the configuration download mode. Then select a new empty directory, such as the MQTTHTTPD directory. To prevent the previous design from remaining, please click the "Remove All" button first, so that the previous design content can be removed. The design file will be saved in this directory and can be downloaded to the device by clicking the "Download" button later.
- Click the "JSON Configuration" button.

ON To	Modbus RTU Settings				>
1. Pe 2. Se	riod of Send to Server: lect the cloud platform to acc	5000	(ms, rang	e: 100 - 31718940	, max 8.8hours)
3. Th GE	e Uplayer Protocol of JSON: T/POST URL(not include the ahe	NONE/MQTT ad "http://")	<u> </u>	_	
Th	e Variable Name of the POST(No	need for pure jso	m):		
4. Ad	d prefix to upload data(eg. Ol	02):		Prefix format	HEX 💌
5. A£	ter 1 times of upload,	serial send data:		Condition(Def.	empty):
6. Ad	d or Remove Modbus Registers:	JSON Upload	JSON Download	Remove All	
7. Cl	ick Save Setting and disply	Save Setting			
8. Ex	port/Import config file.	Upload Export	Upload Import	Download Export	Download Import
					^

(/wiki/File:RS485\_TO\_POE\_ETH06.jpg)

• The parameters here are as follows:

1. **Period of Send to Server:** The default JSON data is sent to the server every time. The server is the destination IP set in the device configuration interface just now, and the unit is milliseconds.

2. **Select the cloud platform to access:** provide functions such as connecting massive devices to the cloud, bidirectional message communication between devices and the cloud, batch device management, remote control, and monitoring, OTA upgrades, device linkage rules, etc., and can flexible transfer device data to Huawei Other cloud services.

3. **The Uplayer Protocol of JSON:** three options (NONE/MQTT, HTTP POST, HTTP GET), choose NONE/MQTT, and do not need to set the domain name, address, and variable name below, choose the other two to fill in according to the actual use.

4. **GET/POST URL:** users can choose whether to add or not, which is mainly used to distinguish devices.

5. **Add prefix to upload data:**This is mainly used to judge a cycle, and this function can be used according to requirements.

6. **Add or remove Modbus registers:** After clicking, you can design the Modbus registers, and you can also view the current content. Delete means deleting all the Modbus registers

designed, which is convenient for restarting the design.

7. **Save setting:** After the design is completed, only click this button to save the data to the download directory just now, and then download it to the device.

8. Export/import EXCELL config file: easy to export and import JSON configuration.

• Now click on the "Add/View" button. For the first row of the previous Modbus table:

Register Address	Parameter	Byte Length	Note
0	Current total power	4	Unsigned, keep 2 decimal places

• The corresponding configuration is as follows:

Following is the [1.	th design of regin	ster. It has been addee	i: F	TOTAL Designer ()	
C keray di	ata(including data by [ ], vithout JS	OS Esyword)	Other Data sou Current Time B	format:	•
Corresponding JSON Reyword:	Data source:	Modbus BIU 💌	Fixed String:		T So quotation
Modbus RTV Settings - Sleve Address: - Nodbus Function Code: - Register Address:		- 645 Protocol (97 vers - 645 Version: 97 V - Device ID: 0000 - Data type: 0410	ion) Version <b>y</b> 1000000001	FE numbers: 0 💌	]
1. Data Length: 2. Decimal Point Flaces: 3. Enable shift and scale:	2 • Bytes. 4 Bytes order: B 0 • digit. After get as inte Subtract integer: 0	ig-Endian(Im▼) (big- nger left shift the de then divide floo	endin 4 bytes: De cimal point. at: 1	ata ABCD. low address	store 2 bytes AB lated
4. Data Format:	Unsigned int . Bool value	at postion bit: 1	-	Enter Esheded	Exit Embeded
				-Design and View Enber Next	Del and Enter
5. Add unit name to rear: 8. Add quotation for data: 7. The Paried Laborary are 977 -	1 100 (as) and bicou the	10		-	

(/wiki/File:RS485\_TO\_POE\_ETH07.jpg)

• The parameters here are described as follows:

1. **The first register:** 1 here means that the current design interface is to configure the first register.

2. **It has been added:** If ticking indicates that it has been added, a tick will appear when viewing the configured information.

3. JSON node data: select object data and array data according to requirements.

4. **Corresponding JSON keyword:** corresponding to the keyword in the uploaded JSON. For example "CurrentW": 232.12.

- 5. Slave address: Modbus table address.
- 6. Modbus function code: currently supports 03 and 04 function codes.
- 7. Register address: the corresponding 0 here.
- 8. Data length: here corresponds to 4 bytes.

9. Decimal point places: 2 digits are reserved here.

10. Enable shift and scale.

11. Data format: here corresponds to an unsigned integer.

12. Add unit name to rear:: according to the data, set the corresponding unit.

13. Add question to data: Yes tick, no tick.

14. Serial port polling time: here is set to 100ms. It refers to the polling interval between this register and the next register, not the polling interval of this instruction.

15 and 16. Function selection, choose according to your needs.

17. Enable next: click to enter the setting of the next register.

18. Del and enter: delete the one that has been added now, and configure the next register.

19. **Save and exit:** After completing the design, click "Save JSON Configuration" on the previous interface.

20. **Cancel and exit:** cancel all current designs, if you want to view the design content, you can click this button to exit.

 Click the "Enable next" button here to continue designing other registers in the Modbus table. After designing all the registers in the table, click "Finish Design", and then click "Save and exit" to exit. Then click the "Download button" on the "Download web" page.

Webpage directly download mode - Webpage directly in local PC:		
MQT	TWQTTHTTPD	•
Special configs:	Clear all	
ZLMB config MQTT config JSON of	config Reg packet	
Code file download mode Select code file:		
C.Virmware.bin		¥
Download through the network Device IP address or domain: Download port (Don't modify):	192.168.1.200     C Download through serial port       1092     Baundrate:	
Device modual/type: 2	003 DevID: 28666CF390A2 Bind ID	
Flash size: 2	56 • кв	
Please close the opened webpage	of the modual in the browser, before start download.	
Stop		

(/wiki/File:RS485\_TO\_POE\_ETH08.jpg)

 Then click "OK" and the device will restart automatically. If it does not restart, please restart manually.

#### **Create MODBUS Analog Meter**

 Here, Modbus Slave is used to analog a table (a serial device is required to simulate Modbus devices, and I use USB TO RS232/485/TTL (https://www.waveshare.com/usb-to-r s232-485-ttl.htm) here).



(/wiki/File:RS485\_TO\_POE\_ETH018.jpg)

 The test results show that the instrument simulated by the Modbus slave tool can be collected by the gateway. At the same time, it can be sent to the server software of the SSCOM simulation according to the JSON format.

# **JSON To MODBUS RTU**

- JSON to Modbus RTU supports 05/06/16 commands. If you need to use the 15 commands to set multiple coils, please use the 05 command multiple times.
- According to the length of the number of bytes, the system will automatically select 06 or 16 commands to send. Here is an example of setting the coil and setting the register respectively.
- If you receive {alert: "on" } JSON data, you need to use the 05 commands to set the station address 02, the coil starting from register 03. Then click "Send JSON" in the JSON to Modbus interface.

SON	N To Modbus RTU Settings					×
1.	Period of Send to Server: Select the cloud platform to acc	1000 ess: None	(ms, range	: 100 - 31718940	), max 8.8hours	)
э.	GET/POST URL(not include the ahe The Variable Name of the POST(No	NONE/MQTT ad "http://") need for pure json):	_			
4.	Add prefix to upload data(eg. Ol	02):		Prefix format	HEX 💌	
5.	After 1 times of upload,	serial send data:		Condition(Def.	empty):	
6.	Add or Remove Modbus Registers:	JSON Upload JSO	W Download	Remove All		
7.	Click Save Setting and disply	Save Setting				
8.	Export/Import config file.	Upload Export Up	load Import	Download Export	Download Impor	rt
Γ						^

(/wiki/File:RS485\_TO\_POE\_ETH019.jpg)

 The configuration interface is as follows: Note that the alert: "on here needs to be written.

When receive data	ert:" on	(including the J	COM name, comma, quo	tation and data)fr	om network.
Then send Modbus write	coil command	with slave address	1 segister	address 3	sontent On 💌
Modbus Write Register C	own and				
When receive data		(including the JS	ON name, command, q	notation) from net	eork.
Then send Modbus write	single/multi	register command v	ith slave address	register a	ldress 0
And the data is follow	ing the JSON	name, the write dat	ta size is 2 💌	Bytes(1 register	is 2 bytes).
The byte order of 4 b	ye is <mark>Big-E</mark> r	ndi an (Invers 💌			
					1.15

(/wiki/File:RS485\_TO\_POE\_ETH020.jpg)

- Click "Next" to add another delivery conversion, otherwise click "Save all and exit". After returning to the main interface, click "Save JSON Settings", and then click "Back". Then pay attention to clicking "Download" on the download interface. This completes the configuration.
- If {power: "12345"} is sent now, the value of power 12345 needs to be set to station address 2, register 3. Then set as follows:

When receive data	(including the JSON name, comma, quotation and data)from network.
Then send Modbus write coil c	command with slave address 1 register address 0 content Off 💌
Modbus Write Register Command	
When receive date power:"	(including the JSON name, command, quotation) from network.
Then send Modbus write single	e/multi register command with slave address <mark>2 register address 3</mark>
And the data is following the	: JSON name, the write data size is 2 🔻 Bytes(1 register is 2 bytes).
The byte order of 4 btye is	Big-Endian (Invers

(/wiki/File:RS485\_TO\_POE\_ETH021.jpg)

Note that you only need to enter power:" for the keyword here, and you don't need to enter the following 12345, because this value changes, but you need to enter a colon. If there are quotation marks in the delivered data, you also need to enter quotation marks.

### MQTT

#### **Device Config**

• First search for the device, then click "Config":

Device Info		Network		Advanced Settings		
Virtual Serial	Not Use 💌	IP Mode	Static	DNS Server IP	8.8.4.	4
Dev Type		IP Address	192 . 168 . 1 . 200	Dest. Mode	Dynamic	
Dev Name	WSDEV0001	Port	4196	Transfer Protocol	None	-
Dev ID	28666CF390A2	Work Mode	TCP Server	Keep Alive Time	60	(s)
Firmware Ver	V1.452	Net Mask	255 . 255 . 255 . 0	Reconnet Time	12	(s)
		Gateway	192 . 168 . 1 . 1	Http Port	80	
Function of the	e device	Dest. IP/Domain	192.168.1.206 Local IP	UDP Group IP	230 . 90 . 76 .	1
DNS South		Dest. Port	4196	Register Pkt:	Г	AS
	d Destocal	Serial		Restart for no d	ata every 300	Se
Modhue Tf	PTORTI	Baud Rate	115200 •	Enable send pa	rameter every 5	Mi
Serial Com	mnad	Data Bits	8 •	More Advaced Settings		
DHCP Sup	port	Parity	None 💌			
Storage Ex	tend	Stop Bits	1 •	Framing Rule Max Frame Length	1300	(By
Multi-TCP	Connection	Flow Control	None 🔹	Max Interval/Smalle	r will better) 3	(Ms

(/wiki/File:RS485\_TO\_POE\_ETH025.jpg)

 Click "Firmware and Configuration" to pop up the configuration download and design dialog box:

on webpageocode domioad	lool		×
Webpage directly download mod Webpage directly in local PC:			
E:V项目VRS485 TO ETH (POE)W	QTTWQTTHTTPD		•
Special configs:	Clear all		
Code file download mode			
EN项目VRS232 TO ETHVRS232	TO POE ETH (B)固件WSDE	V0001.bin	-
<ul> <li>Download through the network</li> <li>Device IP address or domain:</li> <li>Download port (Don't modify):</li> </ul>	192.168.1.200	C Download through serial port Serial port: COM1 → Baundrate: 115200 →	
Device modual/type: Flash size:	2003 256 • K	DevID: 28666CF390A2 Bind ID B	
Please close the opened webp	age of the modual in the bro	wser, before start download.	

(/wiki/File:RS485\_TO\_POE\_ETH026.jpg)

 Here select "Webpage Directory Download", then select an empty directory, such as the MQTTHTTPD directory, and then click MQTT Configuration.

MTT server IP:	broker, emax, io
QTT server port:	1883
Jser name:	
ley:	
QTT ID:	RS485_TO_ETH_(B)
Subscription topic:	mqtt
ublish topic:	mqtt

(/wiki/File:RS485\_TO\_POE\_ETH027.jpg)

- The configuration instructions here are as follows:
- 1. MQTT server IP: fill in here the IP of the MQTT server, up to 30 characters.
- 2. Username: It is the username of the MQTT server.
- 3. Password: It is the login password of this user.
- 4. **MQTT ID:** It is the client ID of MQTT.

5. **Subscription topic:** It is the topic that this device subscribes to. When other devices publish this topic, the server will send it to this device. If you are just publishing, generally you do not need to fill in this field.

6. **Publish topic:** The topic of the data sent to the server when the device converts the serial port to MQTT.

7. Advanced: used to configure advanced parameters.

8. **Save:** Click this button to save the design, and then click the "Download button" in the web download directory to download.

 Now click "MQTT Advanced Settings" (generally no need to configure advanced parameters):

Protocol version:	3.1.1 -	Last-will Retain:	0 💌	ľ
Keep Alive:	60	(s) Will QOS:	0 💌	I
Clean Session:	1	Subscript QOS:	1 💌	I
Enable Will:	0 -	Publish QOS:	1 💌	I
Last-will Topic:		Save Publish:	0 💌	I
Last-will Message				1
		OK	Cancel	1

(/wiki/File:RS485\_TO\_POE\_ETH028.jpg)

Note:

1. **Protocol version:** 3.1.1 is the current popular version, if you want to choose the 3.1 version, you can select it here.

- 2. Keep alive: MQTT heartbeat time, 10 sec at least, 60 sec by default.
- 3. Clean session: whether the server cleans the session after the client is offline.
- 4. Enable will: whether to enable the will.
- 5. Last-will topic: the topic of the last will.
- 6. Last-will message: the message of the last will.
- 7. Last-will retain: whether the server sends the last-will message when the client is offline.
- 8. **Will QOS:** the quality of the will.

9. **Subscript QOS:** the quality of the subscription. Sometimes you need to set it to 0 so as to avoid an offline state when it reloads.

10. **Publish QOS:** the quality of the publishing message from the client. Sometimes you need to set it to 0 so as to avoid an offline state when it reloads.

11. **Save publish:** whether to save the last message in the server. (if there is a new client subscription, it will send it to the client.)

 We do not modify any advanced parameters here, just click: "Save MQTT Config" and then click "Download".

Webpage directly in local PC:	IQTTWQTTHTTPD	•
Special configs:	Clear all	
Code file download mode	A COURT LES Packel	
EN页目VRS232 TO ETHVRS232	TO POE ETH (B)固件WSDEV0001.bin	¥
Download through the network Device IP address or domain: Download port (Don't modify):	Download through serial port           192.168.1.200         Serial port:         COM1           1092         Baundrate:         115200	
Device modual/type: Flash size: Please close the opened webp:	2003   DevID: 28666CF390A2 Bind I 256  KB age of the modual in the proviser before start download	D
Stop		

(/wiki/File:RS485\_TO\_POE\_ETH029.jpg)

 Click OK after downloading, then it will return to the device management dialog. You can see that the device target IP, work mode, and target port have been automatically modified as MQTT settings.

Device	Mana	agement										)
In	Ту	Name	Dev IP	Loc	Dest IP	Work Mode	TCP	Virtual	Vircom St	Dev ID	TX_	
1	Su	WSDEV	192.168.1.200	0	3.228.54.173	TCP Client	Esta	Haven't	Not Linked	6CF390A2	40	
							-					Auto Search

(/wiki/File:RS485\_TO\_POE\_ETH030.jpg)

• If it is not modified automatically, you need to set the target IP, work mode, and target port in the device settings. Then click "Modify Setting".

Device Info		Ne	twork				Advanced Settings				
Virtual Serial	Not Use 💌	IP	Mode	Static		•	DNS Server IP	8.	8.	4 .	4
Dev Type		IP	Address	192 . 168	. 1	. 200	Dest. Mode	Dynam	ic		
Dev Name	WSDEV0001	Po	rt	0			Transfer Protocol	None			
Dev ID	28666CF390A2	r-1 W	ork Mode	TCP Client		•	Keep Alive Time	60			(s)
Firmware Ver	V1.452	Ne	t Mask	255 . 255	. 255	. 0	Reconnet Time	12			(s)
en 11 e 11		Ga	iteway	192 . 168	. 1	. 1	Http Port	80			
Function of the device		De	st. IP/Domain	14.215.190	20	Local IP	UDP Group IP	230 .	90 .	76 .	1
DNS South		De	st. Port	1883			Register Pkt:			- r	AS
	d Destand	Se	Serial				Restart for no data every 300				Sec
Modbus TC	P To PTL	Ba	ud Rate	115200	•		Enable send pa	rameter	every	5	м
Serial Com	mnad	Da	ta Bits	8	-		More Adv	aced Set	tings		1
DHCP Sup	nort	Pa	rity	None	•						
Storage Ex	tend	Ste	op Bits	1	•		Framing Rule Max Frame Length		13	00	10.
Multi-TCP (	Connection	Flo	w Control	None	-		Max Interval(Smalle	r will bet	ter) 3		(Ms

(/wiki/File:RS485\_TO\_POE\_ETH031.jpg)

## **Data Transfer Test**

- The LINK LED of the device will be on (the blue light in the middle) after connecting, which means the device has been connected to the MQTT server normally.
- Connect the UART device and RS485 TO POE ETH (B), and then open SSCOM.

	COM Settings	Display	Send Data	Multi Strings	Tools	Help	联系作者	大虾论坛				
ev Sen	1											
learn	ata OpenFile					an dFila	Stop [7]	arsand T	OnTop Z En	dish Sa	ref on Fi	-] FXT
ConNun ConNun Cl RTS	ata OpenFile COM26 USB Seria SSECon C V DIR BaudBau	l Port More Sett	▼	how <u>SaveData</u> Time and Pack d	Si GverTi	endFile eivedTol ma. 20	Stop Cle File Ser ms No 1 I	arSend厂 adMEX 厂 Se SytesTo 末即	OnTop∣⊽ En ndEvery 50 ह • VerifyN	glish San nz/T lone	reConfig in 🗸 Ad	e ext dCrlf

(/wiki/File:RS485\_TO\_POE\_ETH032.jpg)

 Open the serial port with the same baud rate as the device, and send the data "dev send", and then see the returned data "dev send" in the receiving window. This is because we publish the dev send message to the MQTT server on the MQTT topic. But at the same time, our device also subscribes to the MQTT topic, so the server will immediately send us a subscription message, and the content of the subscription message is dev sent. This information is sent and downloaded as the payload of MQTT, and output from the serial port through transparent transmission. If other devices send information, this device can also receive data.

 Generally speaking, users can directly transmit serial port commands (such as Modbus RTU) directly to the MQTT server. In addition, the JSON function can also be used, which adopts the form of automatic Modbus RTU format collection and regular JSON format upload. In addition, you can also find Waveshare Electronics to customize some nonstandard instruments and host computer protocol formats.

# **MQTT+JSON To MODBUS RTU**

- Combining the above JSON and MQTT can achieve the following functions:
- 1. The MQTT-based protocol is used to establish a connection with the server, and data communication is performed in the form of subscription and publication.
- 2. Support independent design and automatic collection of Modbus RTU registers.
- 3. Support the conversion of specific Modbus register contents into JSON format for regular and active uploading.
- 4. Support adding device ID in JSON format to facilitate cloud identification of devices.
- If you need MQTT + JSON to Modbus RTU function, you can design MQTT and JSON separately, in no particular order. Don' t click the "Clear Design" button after designing one type, and click the "Download" button to download the content to the device after designing both. Generally, you can manually restart the device after downloading it to load the settings.
- Show results:
- MQTT webpage (http://www.emqx.io/online-mqtt-client#/recent\_connections) debug:



Figure MQTT web configuration

JSON setting

ISON	I To Modbus RTU Settings					×
1. 2.	Period of Send to Server: Select the cloud platform to acce	1000 ss: None	(ms, rang	ge: 100 - 31718940	), max 8.8hours)	
3.	The Uplayer Protocol of JSON:	, NONE/MQTT	-			
	The Variable Name of the POST(No	need for pure jzor	ຝ:			
4.	Add prefix to upload data(eg. 01	02):	-	Prefix format	HEX -	
5.	After 1 times of upload,	serial send data:	i	Condition(Def.	empty):	
0.	Add or Kemove Moddus Registers.	JSON Upload	JSON Download	Remove All		
7.	Click Save Setting and disply	Save Setting				
8.	Export/Import config file.	Upload Export	Upload Import	Download Export	Download Impor	t
{ }	"SWV":0.					0
Í						
						×

(/wiki/File:JSON\_setting01.jpg)

MODBUS analog table setting

	□		
	Name	00000	
	0	1	
	1	1	
L	2		(/wiki/File ISON_setting02 ipg)
L	3		(
L	4		
	5		
н	6		
L	7		
	8		
	9		

Communication effect of MQTT + JSON to Modbus RTU



(/wiki/File:JSON\_setting03.jpg)

1

# **HTTP POST/GET+JSON**

 In addition to MQTT, the host computer protocol can also choose HTTP protocol, and upload data through POST and GET instructions. The following takes the POST command as an example to introduce.

ON To Modbus RTU Settings				
1. Period of Send to Server:	1000	(ms, ran	ge: 100 - 31718940	, max 8.8hours)
2. Select the cloud platform to ac	cess: None	•		
3. The Uplayer Protocol of JSON:	HTTP POST	•		
GET/POST URL(not include the ah	ead "http://")	s. com/s	eri/v2.	
The Variable Name of the POST(N	o need for pure jso	n):		
4. Add prefix to upload data(eg. O	1 02):		Prefix format	HEX 💌
5. After 1 times of upload	, serial send data:	[	Condition(Def.	empty):
6. Add or Remove Modbus Registers:	JSON Upload	JSON Download	Remove All	
7. Click Save Setting and disply	Save Setting			
8. Export/Import config file.	Upload Export	Upload Import	Download Export	Download Import
OST /wri/v2 HTTP/1.1 ontent-Type:application/json ontent-Length: lost:s. com				ĺ
"SWV":0.				

(/wiki/File:POST%2BJSON01.jpg)

- The Vircom version adds two options in the JSON to Modbus RTU setting, as shown in the figure:
- 1. Upper-layer protocol of JSON: If it is protocol-free or MQTT protocol, please select the first item: "NONE/MQTT". If it is "HTTP POST", please select the second item "HTTP POST", if it is "HTTP GET", please select the third item "HTTP GET".
- 1. POST/GET URL: When selecting POST or GET, you must fill in the URL. For example, if the URL is http://s.a.com/wri/v2 (http://s.a.com/wri/v2), remove the previous "http://" and directly fill in "s.a.com/wri/v2".
- Other JSON structure design methods are the same as those introduced before, and when you click the "Save JSON Settings" button later, if "POST/GET" is selected, HTTP header format information will be added in front of the JSON data to support the HTTP transmission protocol.
- This "POST/GET" design method is simple and practical, and can easily and quickly transmit instrument data such as Modbus RTU to the server in the form of "HTTP POST/GET + JSON".

Retrieved from "https://www.waveshare.com/w/index.php? title=RS485\_TO\_POE\_ETH\_(B)\_MQTT\_And\_JSON\_User\_Manual&oldid=54065 (https://www.waveshare.com/w/index.php? title=RS485\_TO\_POE\_ETH\_(B)\_MQTT\_And\_JSON\_User\_Manual&oldid=54065)"