

# UART TO ETH User Manual

File version:V1.0



UART TO ETH User Manual .....	1
1. Quick Start.....	5
1.1. Hardware Testing Environment .....	5
1.2. Connection.....	5
1.3. Default Parameter.....	7
1.4. Data Transmission Testing.....	7
2. Overview .....	8
2.1. Brief Introduction.....	8
2.2. Features.....	8
2.3. Parameters .....	9
3. Hardware.....	10
3.1. Hardware Information .....	10
3.1.1. Pin Definition .....	10
3.1.2. Pin Dimension .....	12
3.1.3. Evaluation Kit.....	12
3.2. Hardware Reference Design .....	13
3.2.1. Typical Application Connection .....	13
3.2.2. Power Interface .....	14
3.2.3. UART Interface .....	14
4. Module Function.....	15
4.1 Work Model.....	15
4.1.1. TCP Client Model .....	15
4.1.2. TCP Server Model.....	16
4.1.3. UDP Client Model .....	18
4.1.4. UDP Server Model.....	18
4.1.5. HTTPD Client .....	20
4.1.6. TCP and UDP mechanism .....	20
4.2. DHCP and DNS Function .....	21
4.3. VCOM .....	22
4.3.1. Module Works as Client .....	22
4.3.2. Module Works as Server .....	23
4.3.3. Create VCOM.....	24
4.4. Special Function .....	26
4.4.1. RS485 Function.....	26
4.4.2. Link Function .....	26
4.4.3. Restore factory setting .....	26
4.4.4. Reset Function .....	26
4.4.5. ID Function .....	27
4.4.6. Index Function.....	27
4.4.7. Similar RFC2217 Function .....	27
4.4.8. Firmware Upgrade.....	27
4.5. Additional Function .....	28
4.5.1. AT Command.....	28

4.5.2. Display IP and Data.....	28
4.5.3. Set Client Number in TCP Server Mode .....	29
4.5.4. Defined MAC Address .....	29
4.5.5. Defined DNS Server IP .....	29
4.5.6. Defined Registration Package.....	29
4.5.7. Defined Heartbeat Package .....	29
4.6. Configuration Protocol .....	29
4.6.1. Network setting protocol command.....	29
4.6.2. Network echo command.....	33
4.6.3. Monitor function .....	35
4.6.4. Serial port setting protocol.....	35
4.6.5. Reset command .....	40
4.6.6. Similar T24 series setting protocol .....	40
4.7. UART Framing mechanism .....	42
4.7.1. packing method .....	42
4.7.2. Flow calculation .....	42
5. Parameter Setting .....	43
5.1. Webpage Setting .....	43
5.1.1. Log in.....	43
5.1.2. State Configuration .....	44
5.1.3. Local IP .....	44
5.1.4. Serial Port.....	45
5.1.5. Expand Function.....	45
5.1.6. Misc Configuration.....	46
5.1.7. Reboot .....	47
5.2. Software Setting.....	47
5.3. AT Command .....	51
5.3.1. AT Command Model.....	51
5.3.2. AT Command Set .....	51
5.3.2.1. AT+E.....	51
5.3.2.2. AT +ENTM .....	52
5.3.2.3. AT+Z .....	52
5.3.2.4. AT+VER.....	52
5.3.2.5. AT+MAC .....	52
5.3.2.6. AT+CLEAR .....	52
5.3.2.7. AT+UART .....	53
5.3.2.8. AT SOCKA1 .....	53
5.3.2.9. AT+DHCPEN .....	54
5.3.2.10. AT+WANN .....	54
6. Appendix .....	54
Appendix I :Meaning of Port parameter bit .....	54
Appendix II :ID type of Independent ID .....	55
Appendix III:Upper computer Socket programming example.....	55
Server Socket code: .....	55

Client Socket code: ..... 57

7. Contact information ..... 错误!未定义书签。

8. Disclaimer..... 错误!未定义书签。

9. Undated History ..... 60

# 1. Quick Start

UART TO ETH is used for data bidirectional transparent transmission between TTL and Ethernet. T2 module itself complete protocol conversion, parameter can be set by built-in webpage or software. Once set permanent preservation.

This chapter is quick start for using UART TO ETH module,we advice users to read it carefully and operate personally, it can help you know about module generally.

## 1.1. Hardware Testing Environment

To test T2 conversion function, user should connect T2 UART to computer by USB to TTL serial line, then connect T2 LAN port to computer LAN port by internet cable. If you want to use T2 evaluation board, use USB to RT232 serial line instead of USB to TTL serial line.

Here is schematic diagram for hardware link .



Diagram 1.1-1 Hardware Link

## 1.2. Connection

Computer should be set as follows:

- 1) Shut down firewall and anti-virus software .

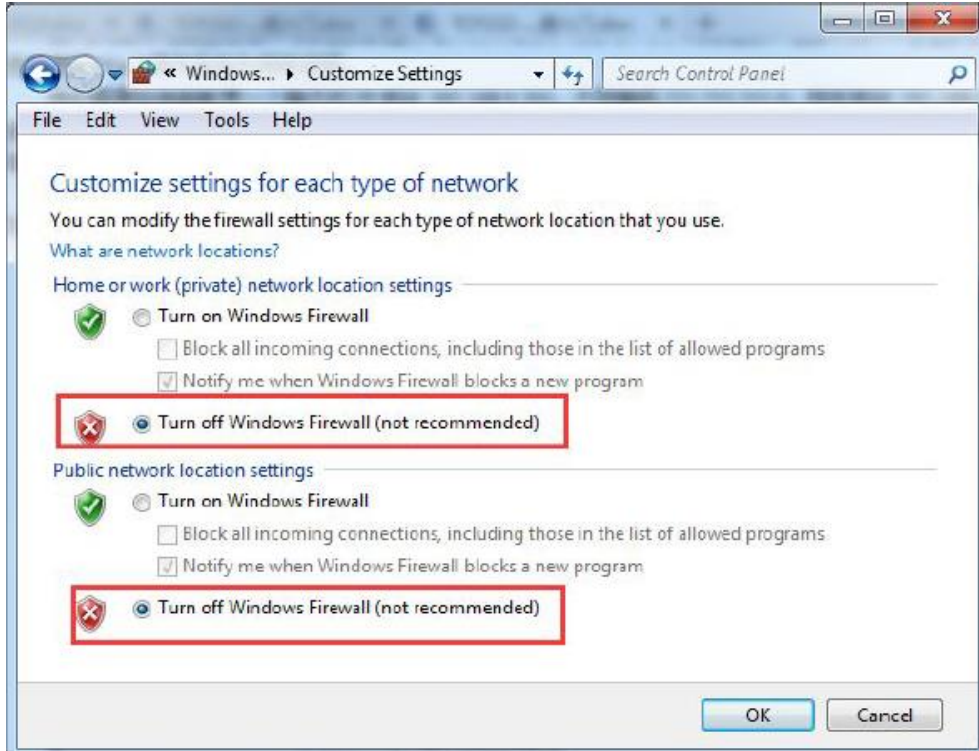


Diagram 1.2-1

2) Shut down unrelated network card, just use one local connection.

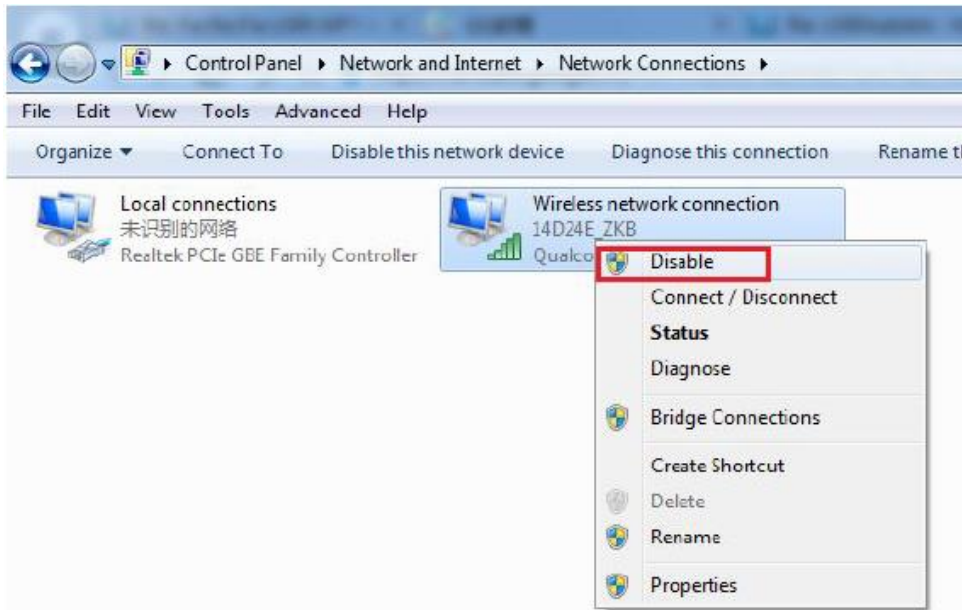


Diagram 1.2-2

3) If you want connect module to PC directly, user should set static IP for computer which is in the same network segment with module.

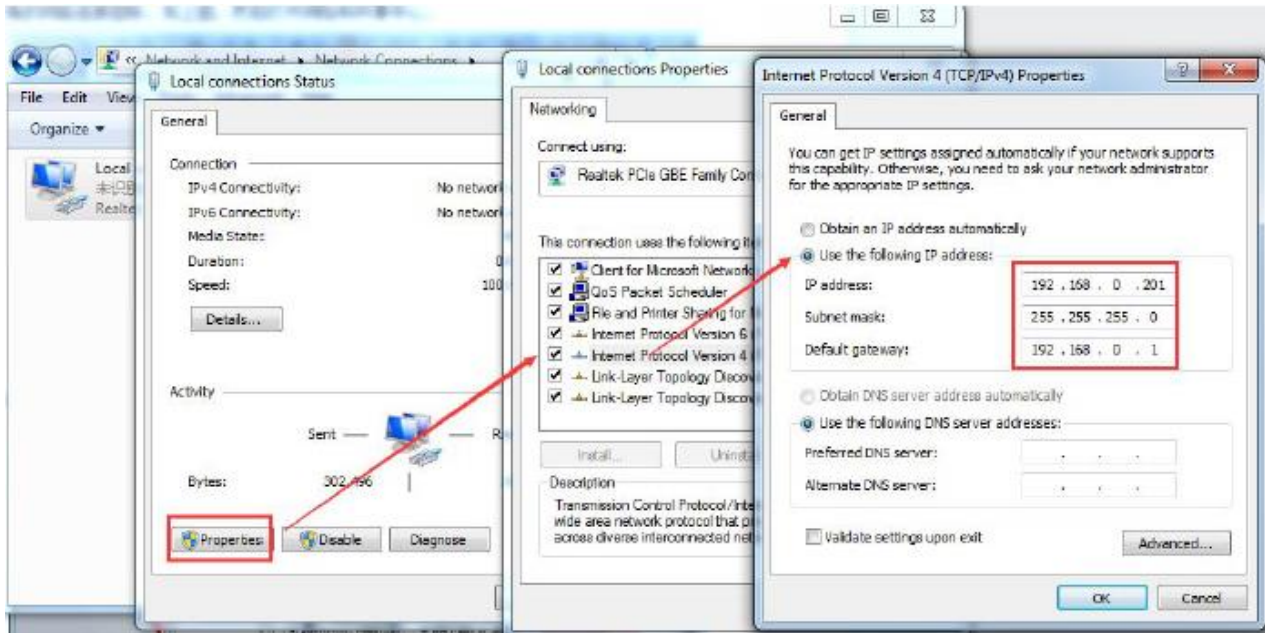


Diagram 1.2-3

### 1.3. Default Parameter

Item	Content
User name	admin
Password	admin
IP address	192.168.0.7
Subnet mask	255.255.255.0
Default gateway	192.168.0.1
Serial baud rate	115200
Serial parameter	None, 8 ,1
Local port	20108
Target IP	192.168.0.201
Target port	8234

Diagram 1.3-1 T2 Module Default Parameter

### 1.4. Data Transmission Testing

Steps for network communication parameters:

- 1) Install USR-TCP232-Test.exe .
- 2) Connect UART to PC, LAN to PC.
- 3) Protocol: TCP Server  
 Server IP: 192.168.201 ( PC Static IP )  
 Server Port No: 8234

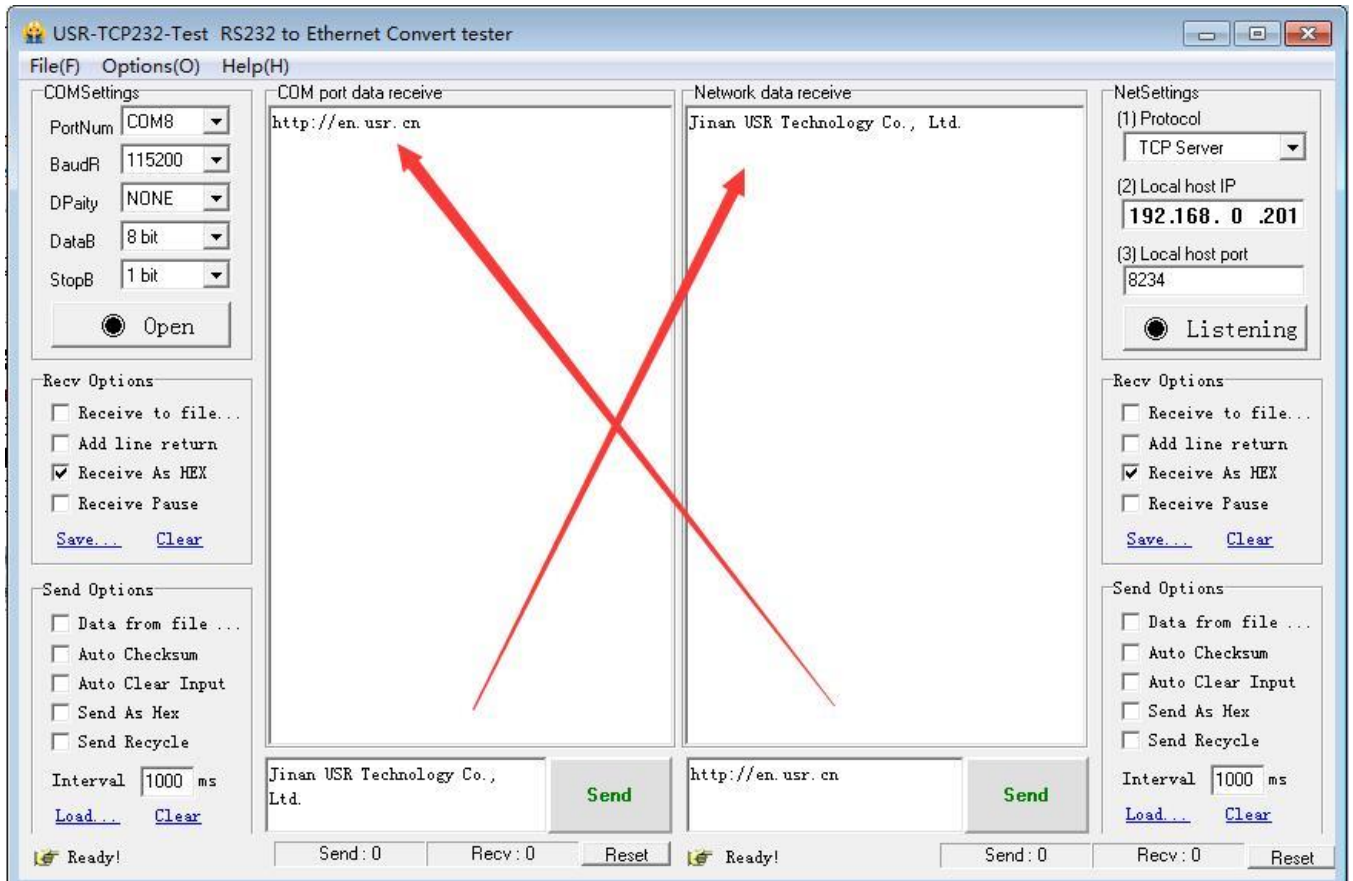


Diagram 1.4-1 Port to LAN Test

## 2. Overview

### 2.1. Brief Introduction

TCP232-T2 is a new and tiny size serial to Ethernet module which realizes data bidirectional transparent transmission between TTL Port and RJ45 Port, it can also used in RT232/ RS485 by level shift circuit.

T2 is equipped with Cortex-M0 core. It has characters of low power, fast speed, high efficiency, strong compatibility, it is easy to use.

### 2.2. Features

- Support DHCP (Dynamic Host Configuration Protocol);
- Support DNS (Domain Name System);
- Web-set: Setting parameters through web;
- Upgrade firmware via network;
- Support AUTO MDI/MDIX, can use a crossover cable or parallel cable connection;
- Serial port baud rate 600 bps ~460.8Kbps, and None, Odd, Even, Mark, Space, five check bits;
- Work mode: TCP Server, TCP Client, UDP Client, UDP Server, HTTPD Client;



- Working model related parameters can be set via a serial port or network;
- Support virtual serial port, self-developed USR-VCOM software;
- Heartbeat package mechanism to ensure connection is reliable, put an end to connect feign death;
- User-defined registration package mechanism, check the status of connection;
- Under TCP Server model, Client number range from 1 to 16, default value is 4;
- Support User-defined MAC address;
- Restore factory default;
- Across the gateway, switches, routers;
- Across the gateway, across switches, routers;
- Provide(socket), VB, C++, Delphi, Android, IOS;
- Download application cases;
- Support customization;

### 2.3. Parameters

Parameter	Parameter Value
Voltage	VCC: DC 3.3V, 3.15V~3.45 V VDD: DC 5V, 4.75V~5.5V
Current	130mA
Consumption	<1W
Serial Level	TTL
LAN Port	RJ45 with electromagnetism isolation
Packing	DIP
Size	PCB: 50.5*22.6mm(L*W) Module: 55.0*22.6*23.1mm (L*W*H)
Temperature	Working temp: -25 ~ 75 °C Storage temp: -40 ~ 105 °C Storage humidity: 5% ~ 95% RH

Diagram 2.3-1 UART TO ETH Parameters

### 3. Hardware

#### 3.1. Hardware Information

##### 3.1.1. Pin Definition

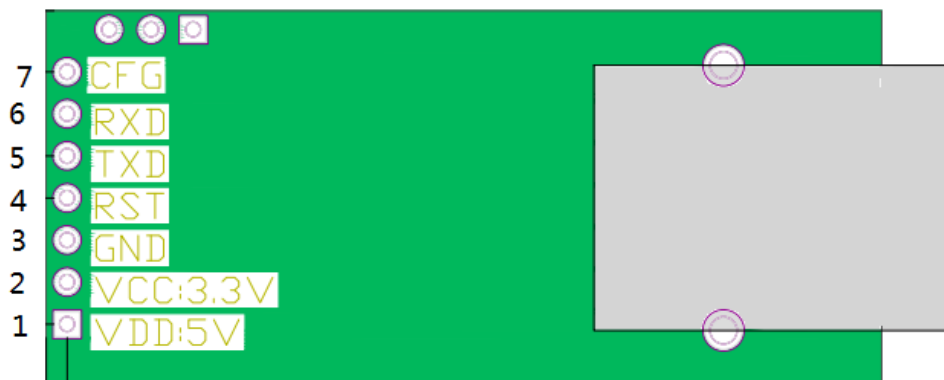


Diagram 3.1.1-1 T2 Interface Definition

NO	Pin	Function	Description
1	VDD	Power supply	Typical value 5V @ 200mA
2	VCC	Power supply	Typical value 3.3V @ 200mA
3	GND	Ground signal	Connect to ground
4	RST	Restart	When it receives 200ms low level , module will restart. It's advised that in any case , MCU controls the pin by pull up resistor.
5	TXD	Send data	TTL connect to 3.3v MCU (For 5V, refer to Diagram 3.1.1-2)
6	RXD	Receive data	TTL connect to 3.3v MUC (For 5V, refer to Diagram 3.1.1-2)
7	CFG (Reload)	Restore factory default	Pull down this pin for 5s , module will reset to factory setting. the pin needs to connect 10K pull up resistor to 3.3V. <b>Refer to 4.4.3 Restore factory setting</b>

Form 3.1.1-1 T2 Pin Definition

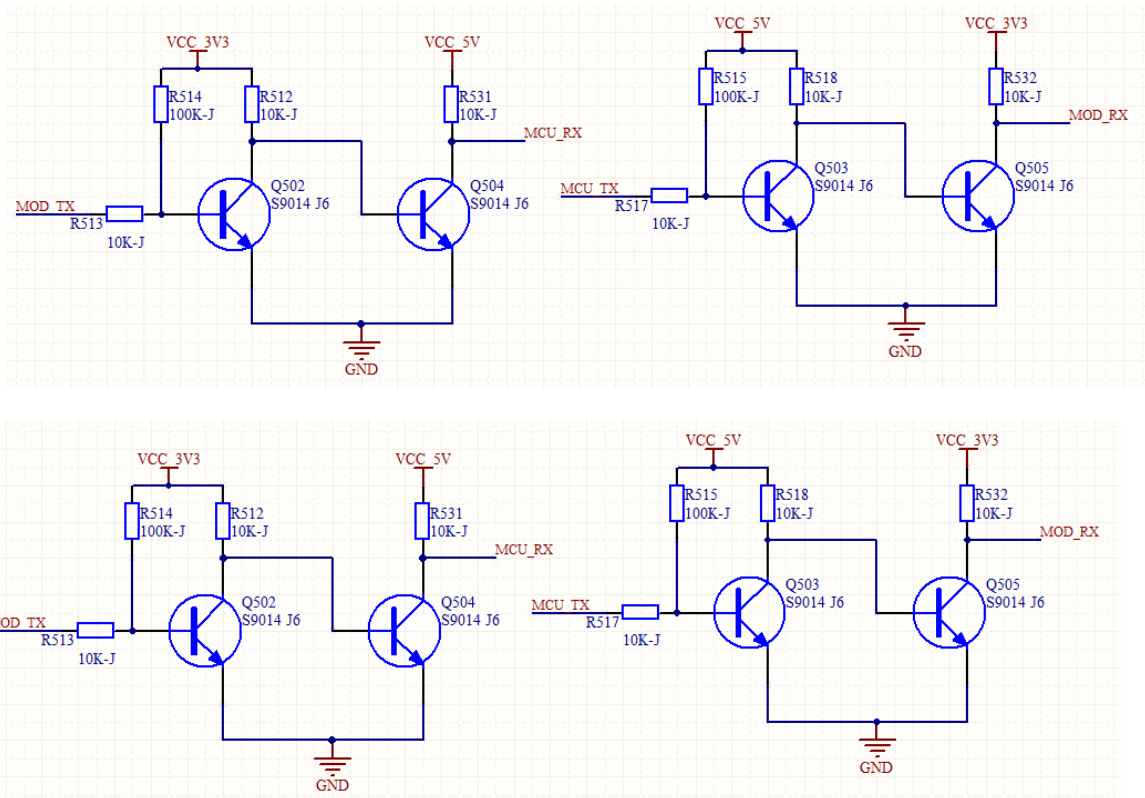


Diagram 3.1.1-2 3.3V to 5V voltage conversion circuit

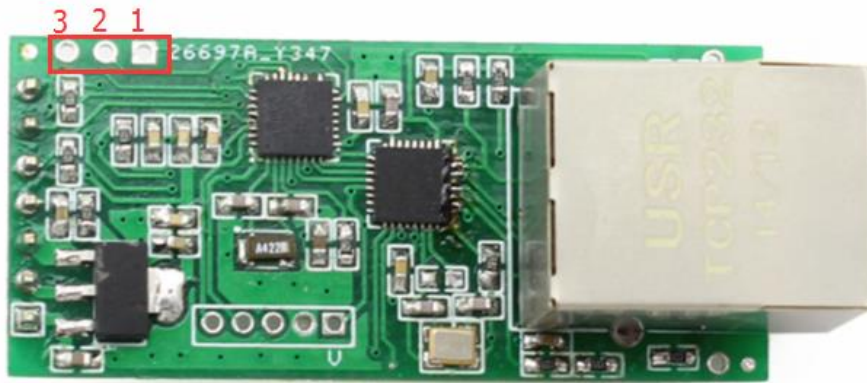


Diagram 3.1.1-3 Reserved Pin

1	485_en	Reserved	RS485 reserved pin
2	Link	Reserved	Used as indication pin for TCP connection status. <b>Refer to 4.4.2 Link Function</b>
3	ISP	Reserved	Hasn't use

Form 3.1.1-2 T2 Reserved Pin Definition

### 3.1.2. Pin Dimension

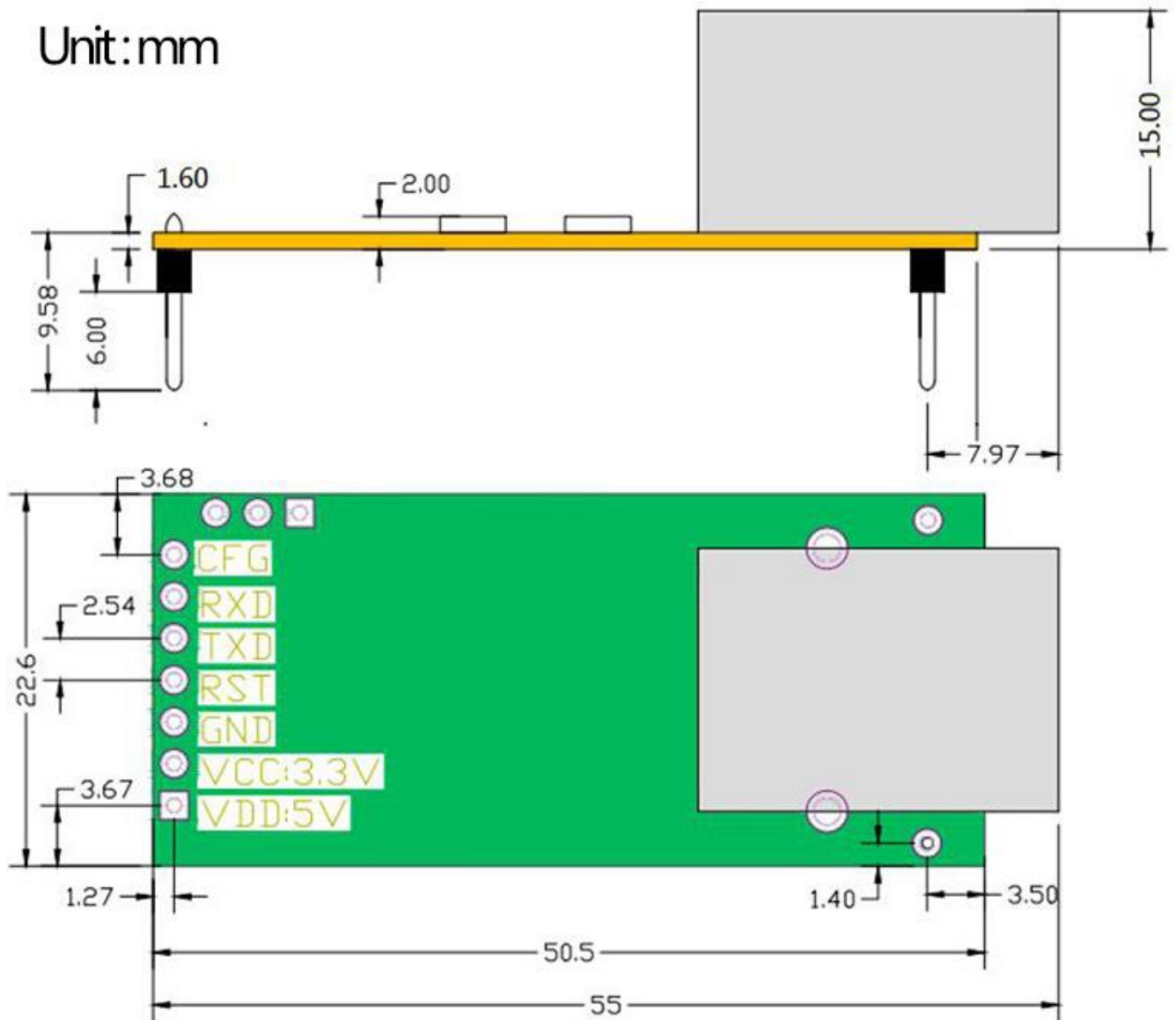


Diagram 3.1.2-1 T2 Dimension

### 3.1.3. Evaluation Kit

USR-TCP232-EVK evaluation board can be used for TCP232-T2

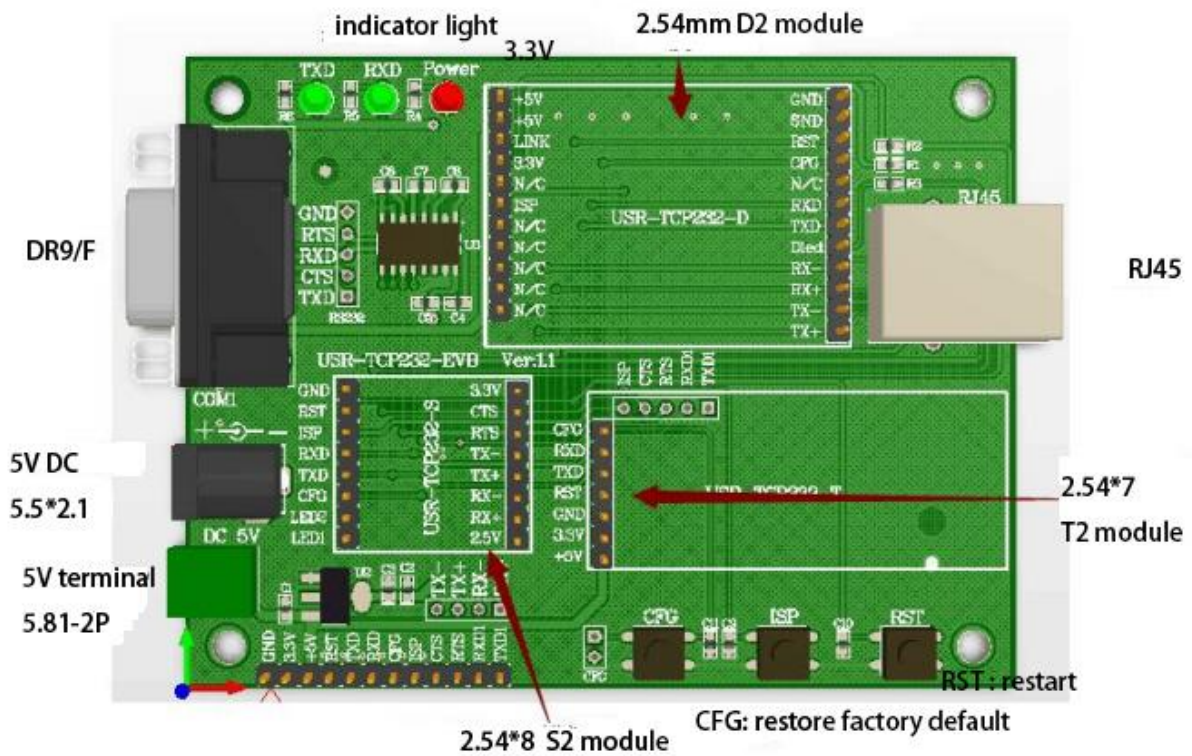
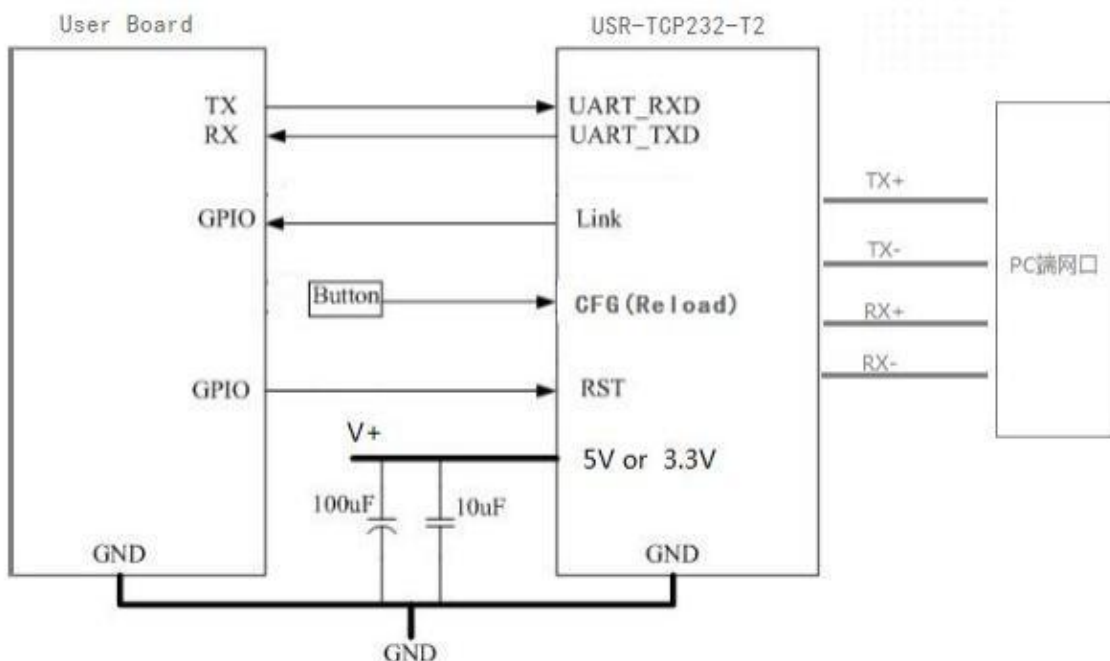


Diagram 3.1.3-1 Evaluation Kit

## 3.2. Hardware Reference Design

### 3.2.1. Typical Application Connection



**Diagram 3.2.1-1 Typical Application Connection**

Note:

- 1) RST: Restart button. It works at low level, 3.3V, 10K pull-up resistor inside. When it access to power or breakdown, pull down 300ms then high level.
- 2) Link: connection indication.Can be used as indication pin for net connection, when connected, it outputs low level; When unconnected, it output high level.
- 3) CFG(Reload): connect to external button or configuration pin. 3.3V, 10K pull-up resistor inside. Press it and pull to low level, then release hands after 3 seconds, module restores factory default and restart.
- 4) UART\_TXD/RXD: data send/ receive signal(10K pull-up resistor inside)

**3.2.2. Power Interface**

UART TO ETH has dual power supply interface, 5.5 v and 3.3 v (only choose one)

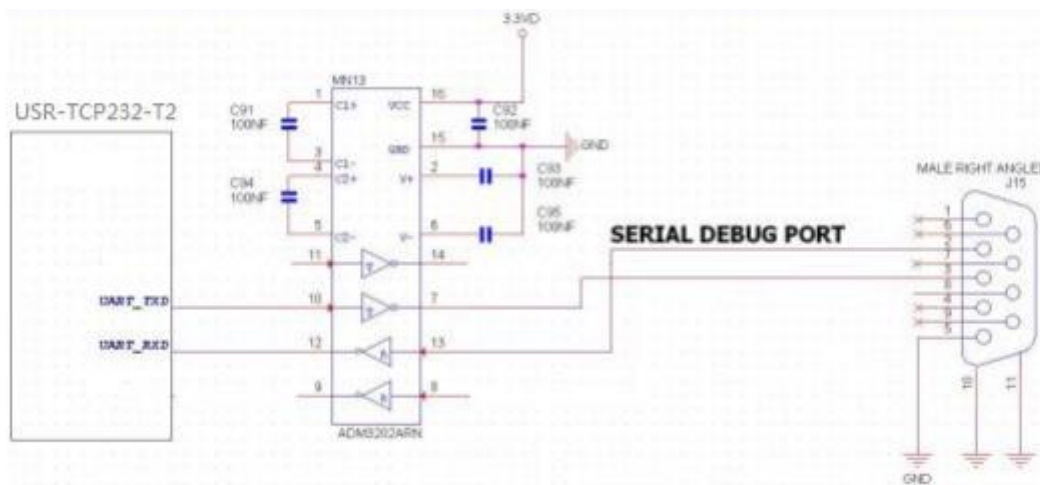
Working current 150 mA, max current 200mA .

VCC: typical value DC 3.3V, 3.15V~3.45 V. It is used for 3.3 V MCU.

VDD: typical value DC 5V, 4.75V~5.5V. When connect to 5V MUC, **refer to Diagram 3.2.3 -2 3.3V to 5V voltage conversion circuit**

**3.2.3. UART Interface**

UART can connect RS232 chip, UART has TXD/RXD signal wire. Take RS232 Level for example.



**Diagram 3.2.3-1 UART Interface Design**

If communicated with MCU (3.3V TTL), connect module's TXD to MCU'S RXD, modules' RXD to MCU'S TXD. If MCU 5V TTL, need voltage conversion circuit, as follows:

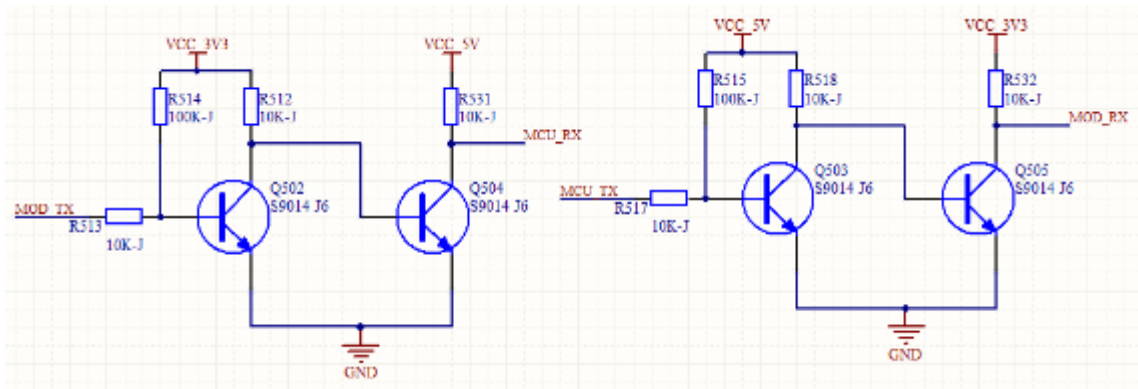


Diagram 3.2.3-2 UART Level Switch Design

## 4. Module Function

### 4.1 Work Model

#### 4.1.1. TCP Client Model

It has to be connected before transferring data.

- 1) In TCP Client Mode, TCP232-T2 connects TCP Server actively, establish a connection to transmit data
- 2) In TCP Client Mode, It has function of identifying disconnected link. When connected, it will send keepalive package every 15s. If unconnected, it can be detected timely and enforce TCP232-T2 to disconnect the former link to establish a new one.
- 3) When TCP232-T2 try to connect remote server, if the local port number is not "0", it will establish a connection with the same source port every time.
- 4) It has synchronizing function of baud rate, user should install USR VCOM Software.
- 5) When local port number is "0", it means local port is random.



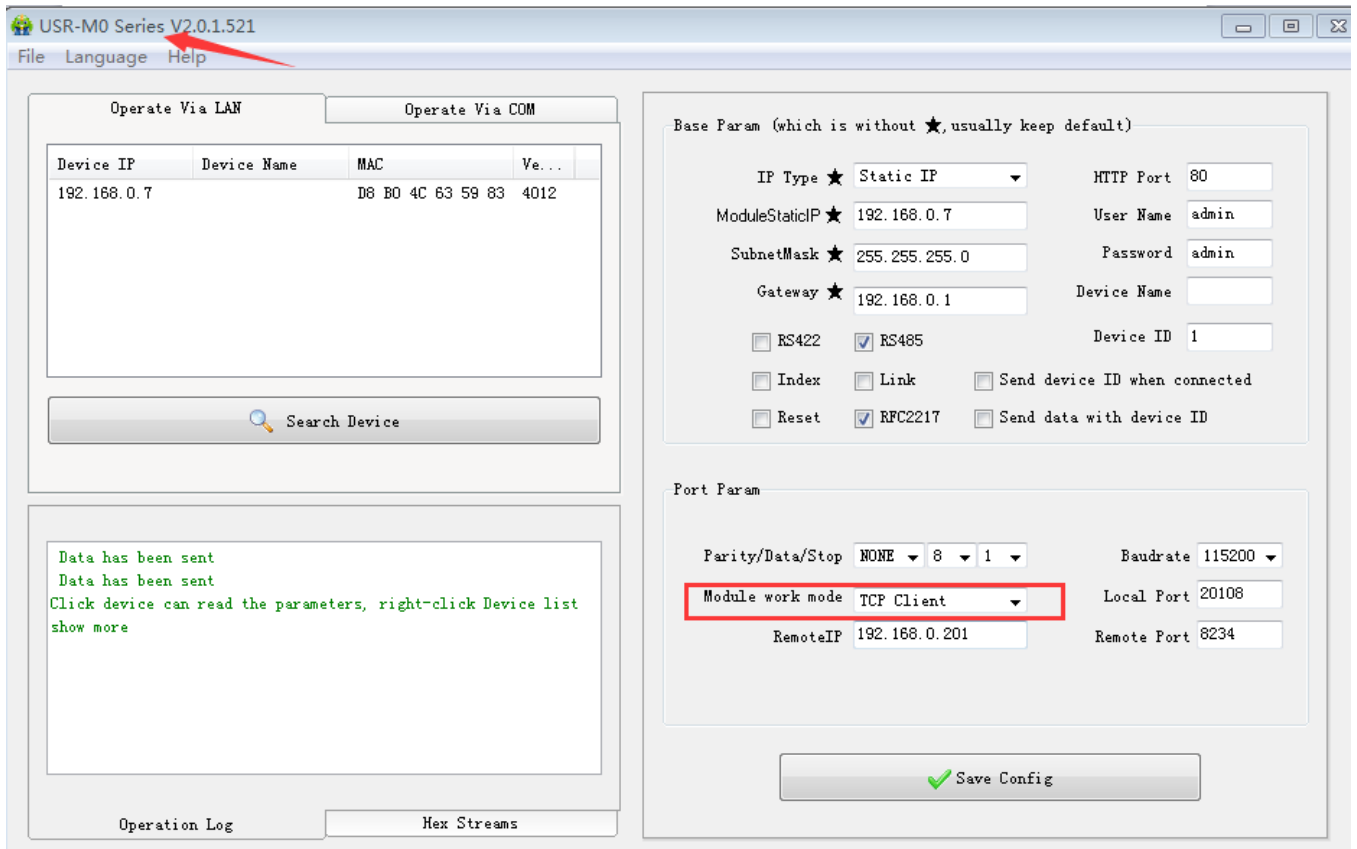


Diagram 4.1.1-1 TCP Client Setting

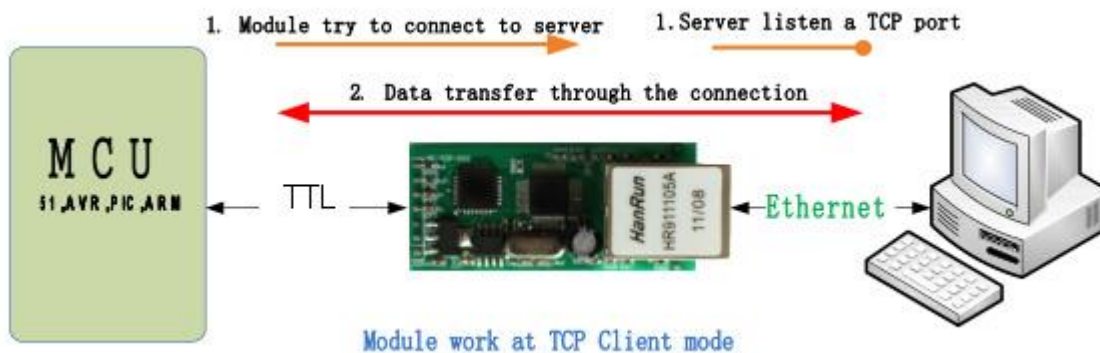


Diagram 4.1.1-2 TCP Client Model

### 4.1.2. TCP Server Model

- 1) It has to be connected before transferring data.
- 2) In TCP Server Mode, T2 monitors local port, it will response and establish a connection when there is a request. Up to 4 links at the same time. Once received data, T2 serial port will send data to all the devices which connect to TCP232-T2.
- 3) It has synchronizing function of baud rate, user should install USR VCOM Software.
- 4) In TCP Server Mode, The maximum number can be configured by user. TCP Client number is from 1 to 16, default value 4. When the Client link is more than 16, the new link will replace the former link from Link 1. If the Client link is more than 4, send and receive data at same time , the data flow should be within 2.5 KB/s



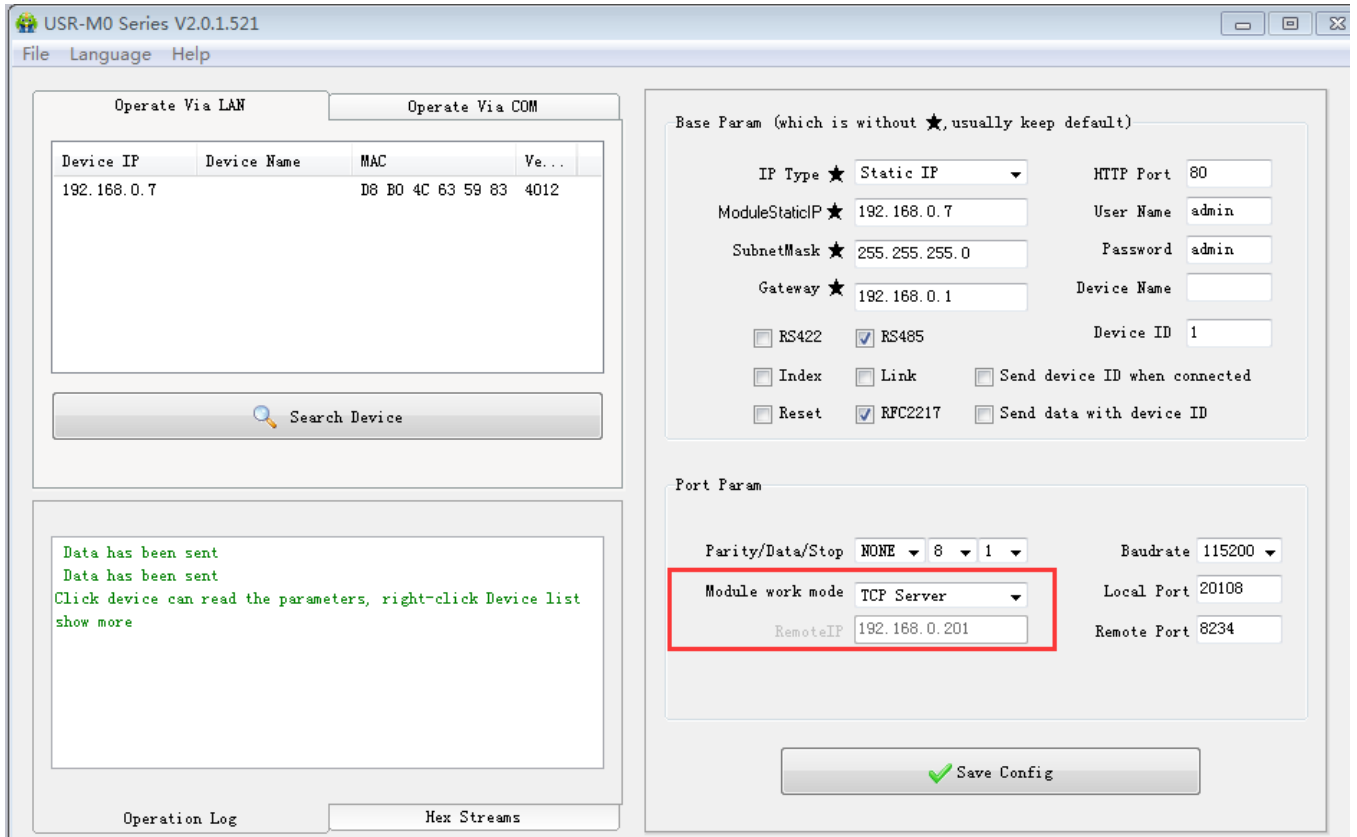
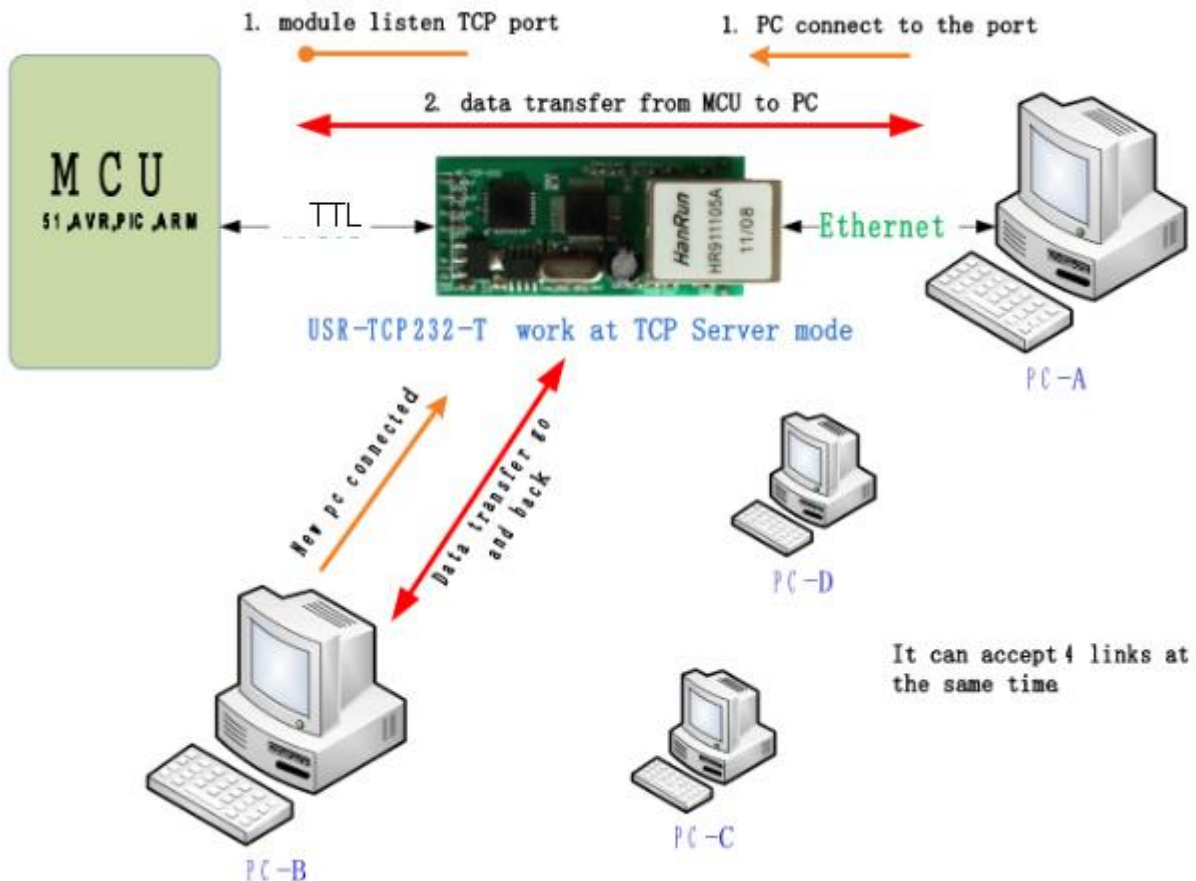


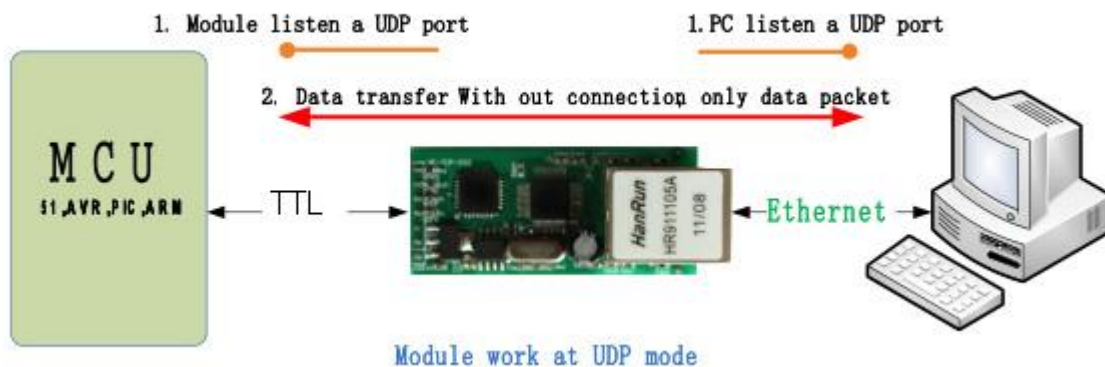
Diagram 4.1.2-1 TCP Server Setting



**Diagram 4.1.2-2 TCP Server Model**

**4.1.3. UDP Client Model**

- 1) The Model belongs to UDP Protocol.
- 2) In UDP Client Mode, TCP232-T2 won't establish the connection actively. It can only communicate with the target port whose IP has been set. When serial port receive data, it send data to target IP and port. If data doesn't come from this channel, it will not be accepted by TCP232-T2.
- 3) In UDP Client Mode, if target IP is set as 255.255.255.255, it can realize function of entire network broadcast, also can receive broadcast data. If broadcast in network segment ,eg.192.168.0.255, it can only send data ,can't receive data.
- 4) Under UDP Client, maximum data length sent from MCU to TCP232-T2 is 1460.



**Diagram 4.1.3-1 UDP Client Model**

**4.1.4. UDP Server Model**

- 1) UDP Server is based on normal UDP, it doesn't validate the source of IP address. Once received UDP data, it convert target IP to data source IP, similar to TCP Server.
- 2) In UDP Server Mode, TCP232-T2 records an IP, Once it receives data, it will send data to record IP. TCP232-T2 also works as a server, can receive data from Ethernet and convert target IP to data source IP.

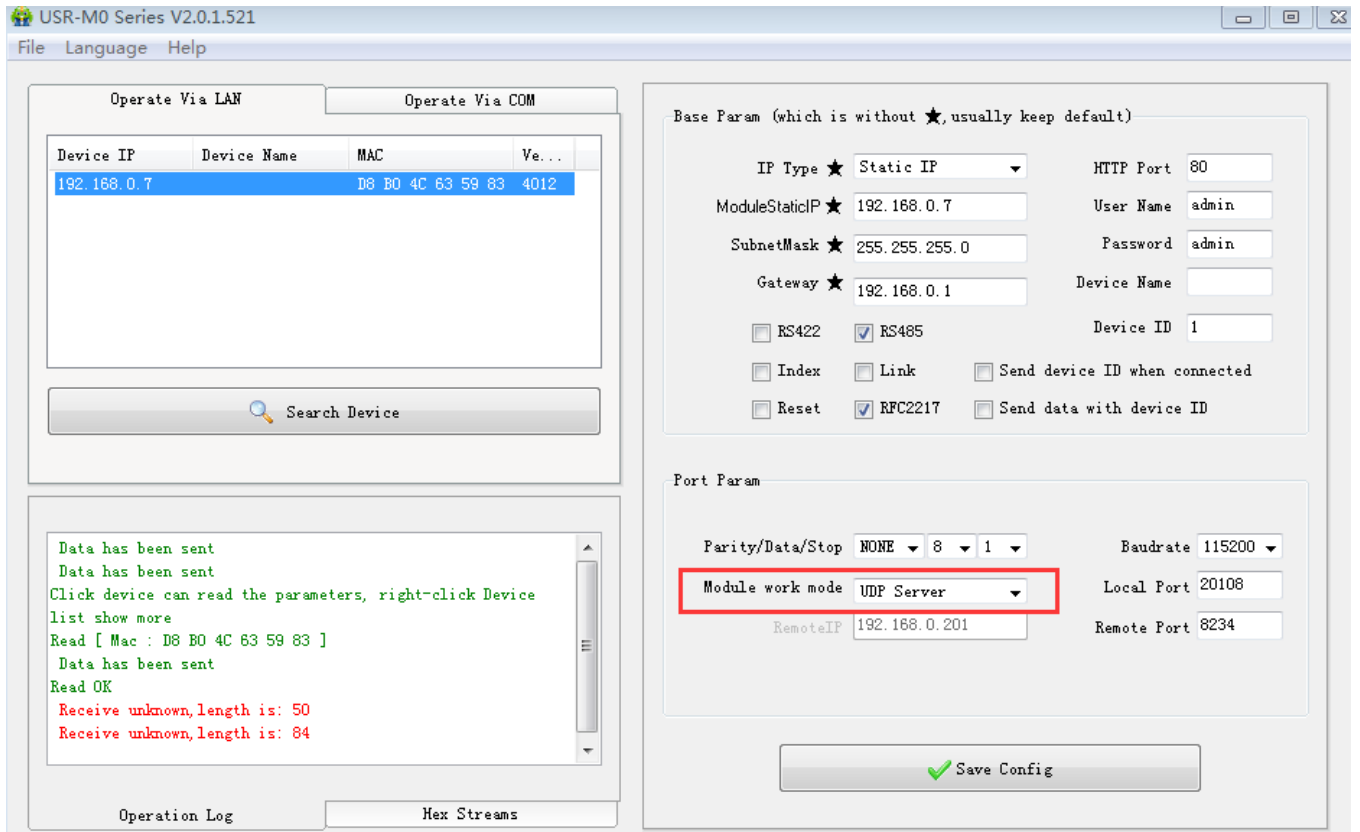


Diagram 4.1.4-1 UDP Server Setting

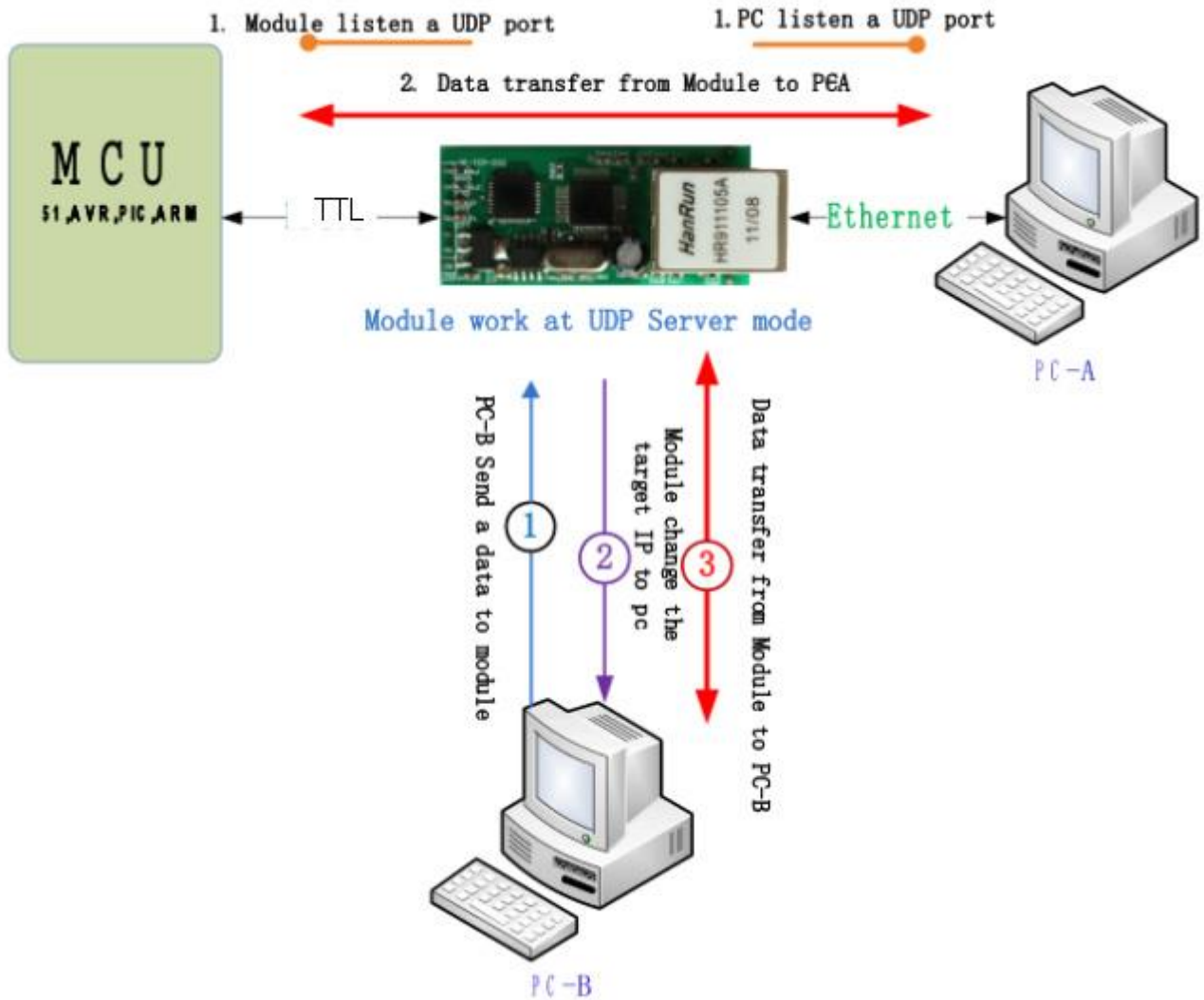


Diagram 4.1.4-2 UDP Server Model

#### 4.1.5. HTTPD Client

This function is used for developer.

- 1) Module T2 send data to HTTP Server or receive from HTTP Server, complex HTTP protocol will be done by T2, it is convenient for user to programming.
- 2) T2 received data from HTTP Server will send to serial port without process.
- 3) According to demand, user can define HTTP content.

#### 4.1.6. TCP and UDP mechanism

	TCP	UDP
Advantages	Stable; Not easy to lose data package; Reliable connection mechanism;	Transmission interval is accurate; No connection mechanism; Easy and flexible;
Disadvantage	Easy to block up Information; Because of check and resend mechanism, interval isn't accurate	Under bad network condition, it is high risky to losing data package

## 4.2. DHCP and DNS Function

DHCP: Dynamic Host Configuration Protocol

When T2 connects to remote server, it can obtain an IP address automatically which router or gateway distributed. If you don't know how to set IP address or it can't connect because of the set IP is not in the same segment, the function is helpful for you.

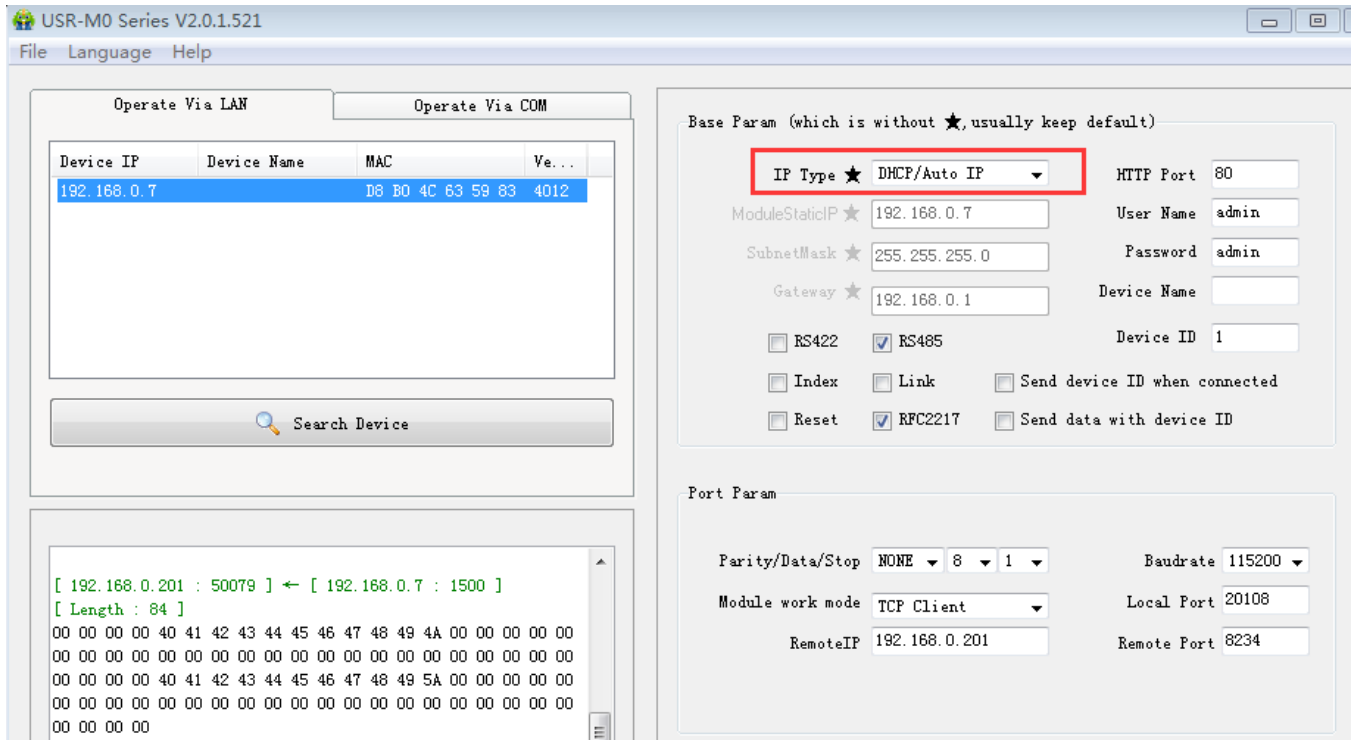


Diagram 4.2-1 DHCP

DNS: Domain Name System

e.g. domain name sever is cloud usr.cn, when we don't know Server IP or Server IP changed, this function plays an important role.

Note: when use NDS function, T2 gateway must be same as router IP or choose DHCP function.

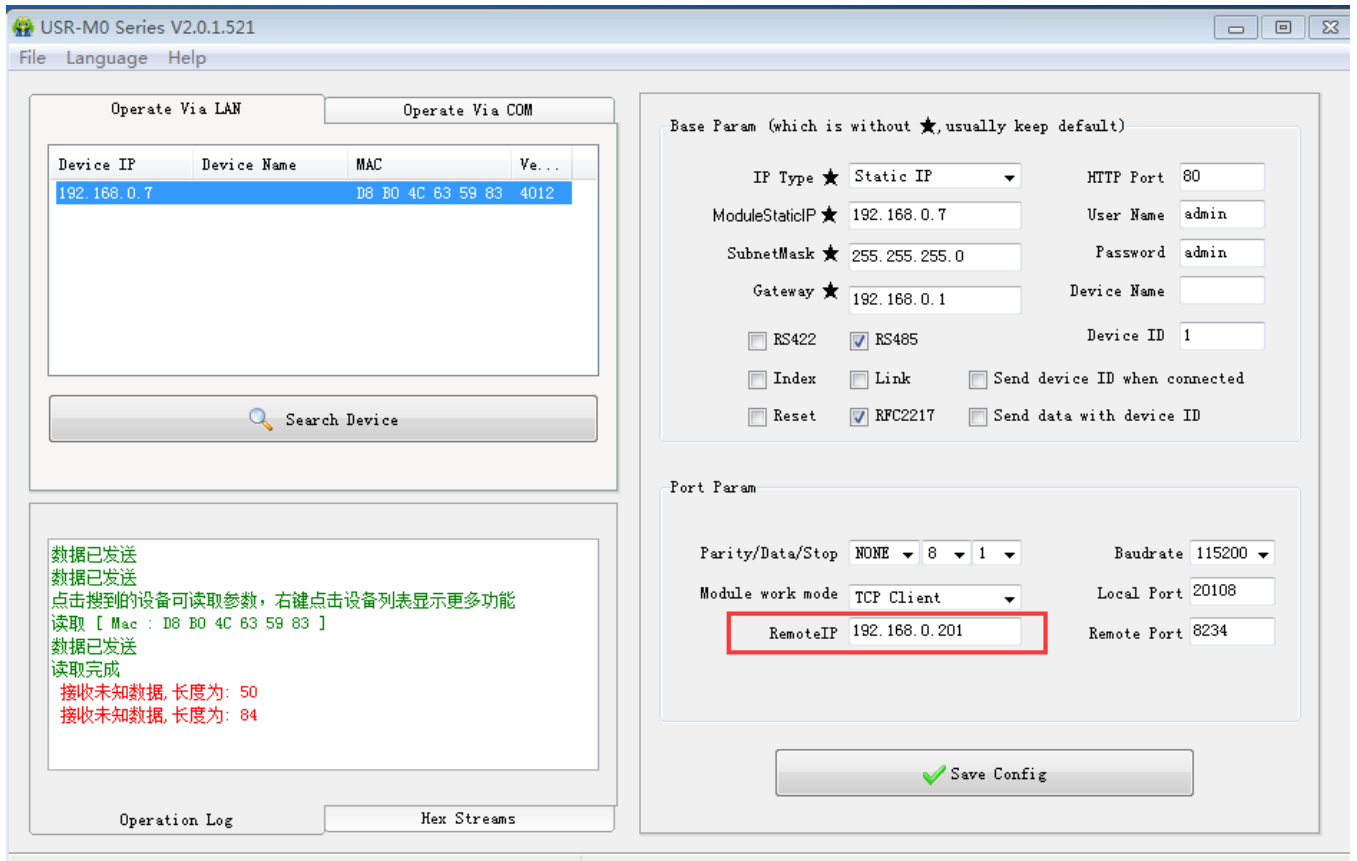


Diagram 4.2-2 DNS

## 4.3. VCOM

USR-VCOM Download: [http://www.waveshare.com/wiki/UART\\_TO\\_ETH\\_Software](http://www.waveshare.com/wiki/UART_TO_ETH_Software)

If user's upper computer and device are all connect by serial port, user can create a COM which has TCP/IP to realize remote control by USR-VCOM software.

- 1) Turn off firewall and anti-virus software.
- 2) Install USR-VCOM.

I advise user to choose "Search" or "Smart vcom" to create virtual port. **Please refer to 4.3.3**

### 4.3.1. Module Works as Client

- 1) Set module parameters. T2 work model: TCP Client.
- 2) Open USR-VCOM, set virtual port as follows :

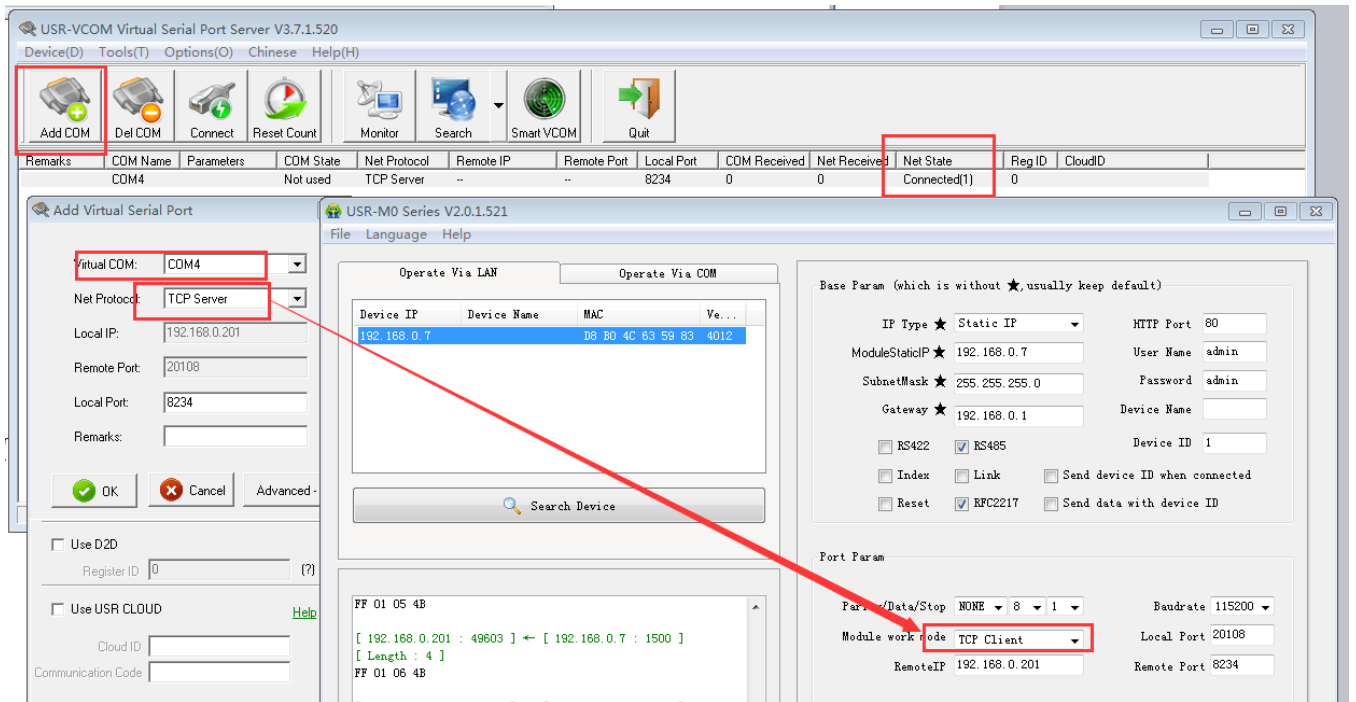


Diagram 4.3.1-1 Create Server Virtual Port

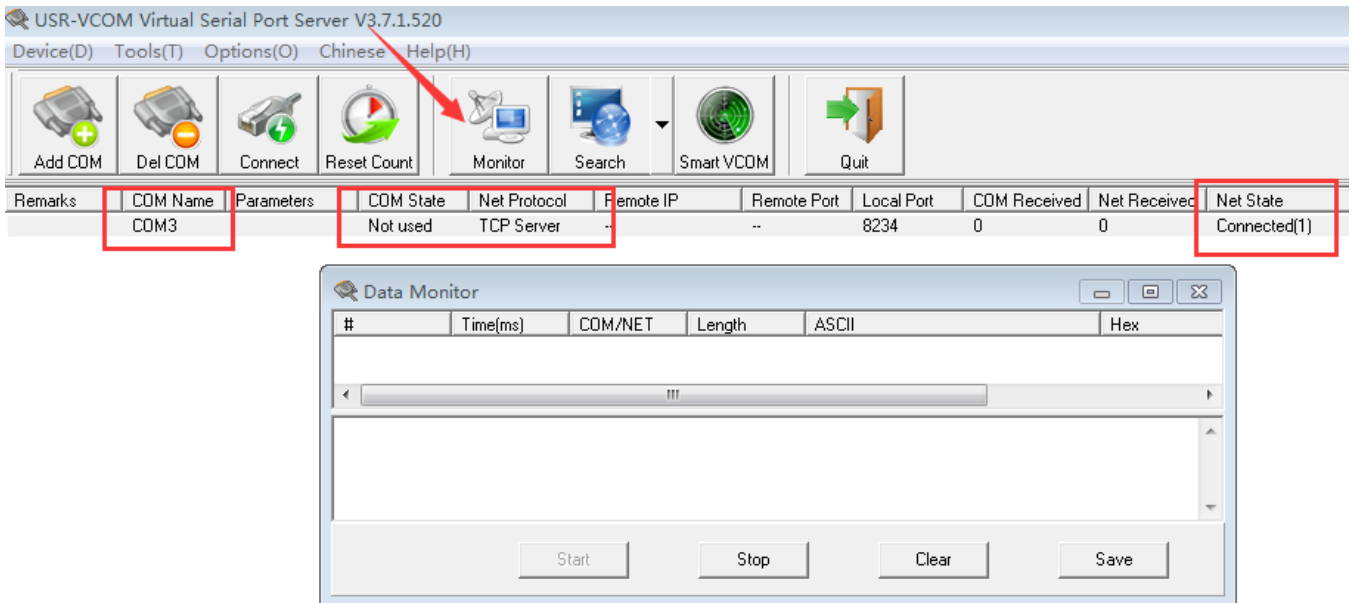


Diagram 4.3.1-2 Monitor Date

### 4.3.2. Module Works as Server

- 1) Set T2 work model: TCP Server.
- 2) Set virtual port as follows:

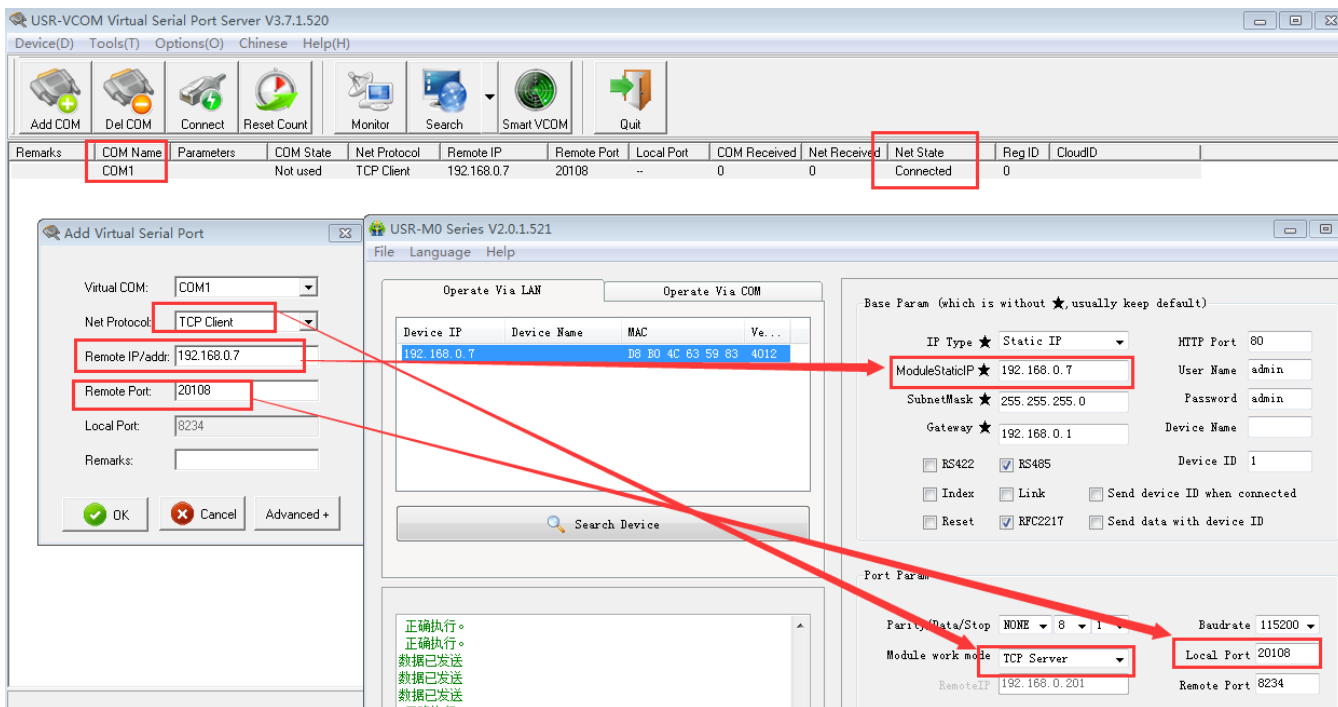


Diagram 4.3.2-1 Create Client Virtual Port

### 4.3.3. Create VCOM

1) Create VCOM by “search” button.



Diagram 4.3.3-1 Search Function



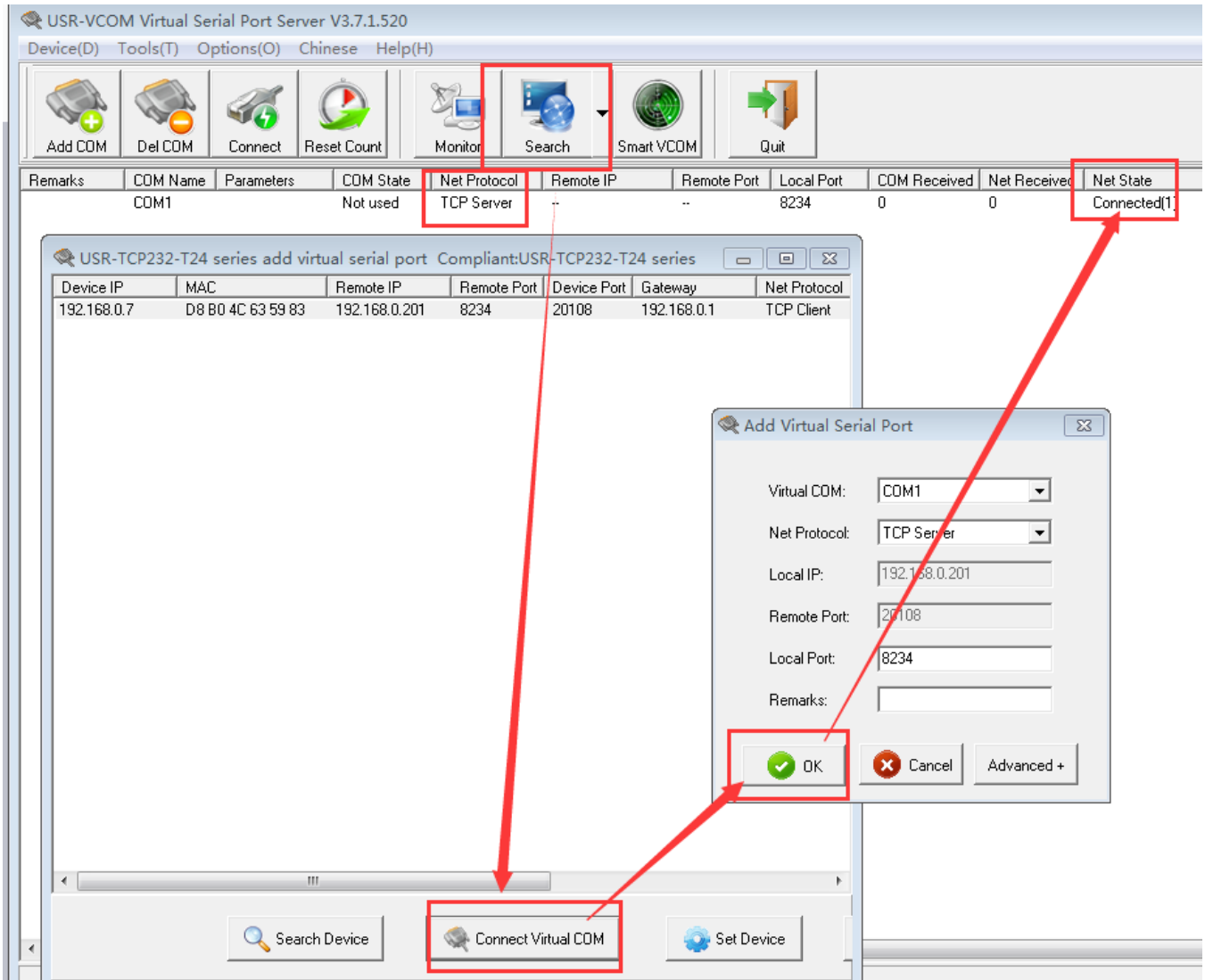


Diagram 4.3.3-2 Search Function

2) Automatically Create

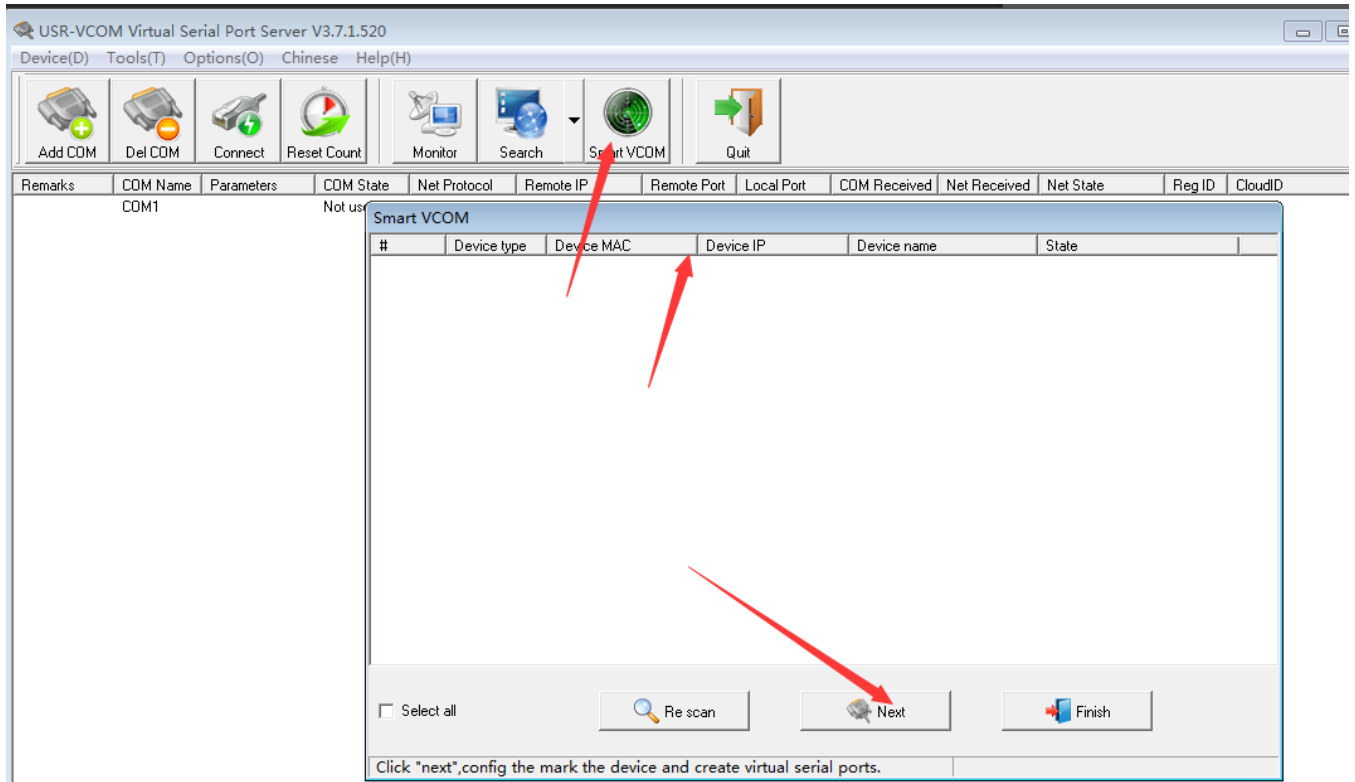


Diagram 4.3.3-1 Smart VCOM

## 4.4. Special Function

### 4.4.1. RS485 Function

T2 reserved 485 pin, this function can be set by software, won't effect RT232 communication.

### 4.4.2. Link Function

Link pin can be used as indication pin for TCP connection status.

When connected, it output low level; When unconnected, high level.

When T2 is under TCP model, Link pin will pull down. otherwise, it stays in high level.

When T2 is under DUP model, Link pin will always pull down. By default ,it is not checked .

### 4.4.3. Restore factory setting

1) 1) Hardware: At first, check "Reload" in webpage or software, then Reload can only used to restore factory settings. After setting then module will reset, pull " Reload " down to 0 V level for 5 seconds then pull CFG( Reload) up to 3.3 V or don't connect it, factory reset is finished.

2) Software: finish it by set-up software.

3) AT Command: After entering into AT Command, then send AT+ clear.

### 4.4.4. Reset Function

When T2 works as TCP Client, T2 connects to TCP Server actively. When start Reset function, T2 try to

connect to TCP Server for 30 times. If failed, T2 will restart automatically. By default, it won't be chose.

#### **4.4.5. ID Function**

When T2 works as TCP Client, T2 connects to TCP Server actively. When start Reset function, T2 try to connect to TCP Server for 30 times. If failed, T2 will restart automatically. By default, it won't be chose.

#### **4.4.6. Index Function**

When T2 works as TCP Server, it can establish 16 links simultaneously at most. Default is 4 .The max number can be set from 1 to 16. Take 4 link as example, Server send data to 4 Client simultaneously or Server can't distinguish the data source, Index can realize the choice of data source of sending or receiving.

Index function can be set by software or web-page.

#### **4.4.7. Similar RFC2217 Function**

This function is used to change UART TO ETH serial port parameter through network.

E.g: change baud rate from 115200bps to 9600bps. It can be set by software or webpage.

By default ,it is in open state.

When using VCOM software, this function also works, the baud rate of software in PC will match Autonomic with the baud rate of UART TO ETH, don't need to focus on serial port baud rate. After restart, it is default parameter .

#### **4.4.8. Firmware Upgrade**

When Module T2 IP and PC IP is in the same segment , click here, then click "firmware upgrade"

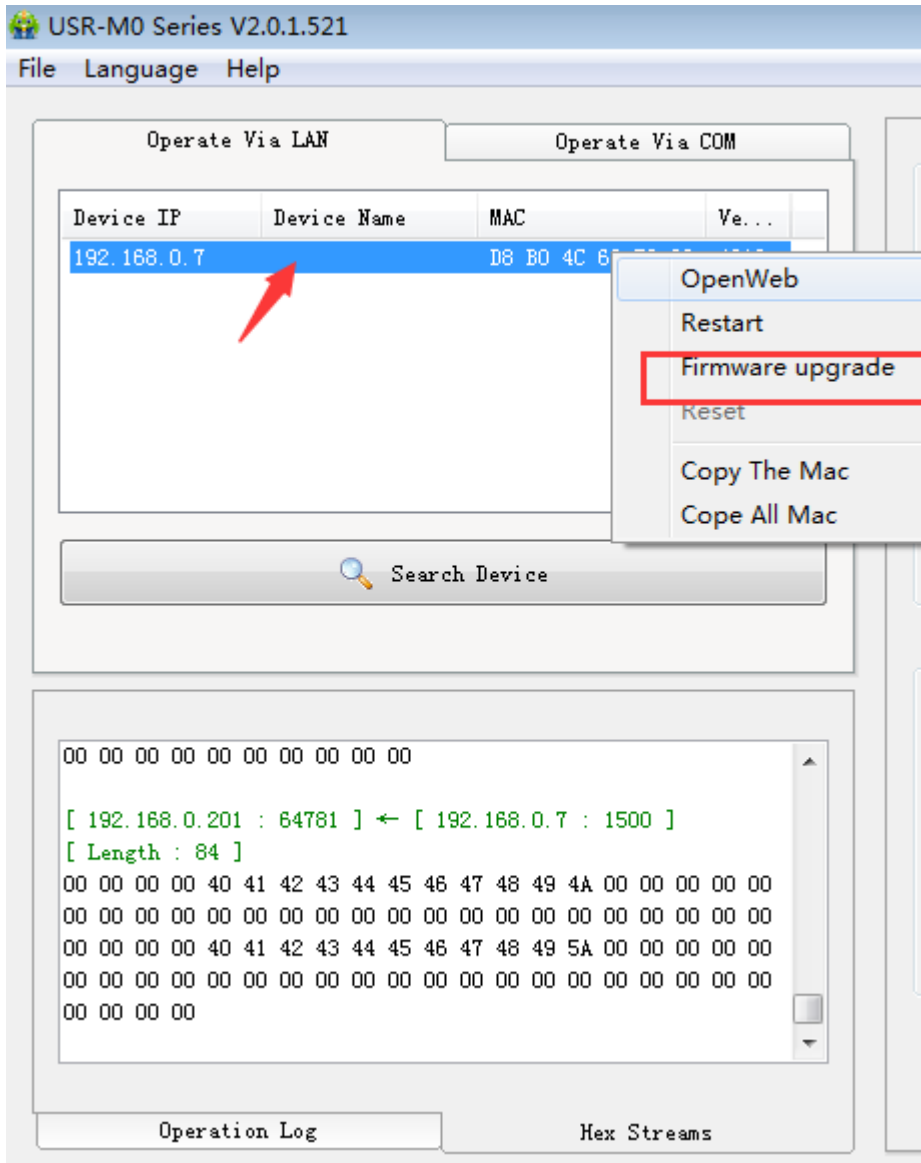


Diagram 4.1.7-1 Firmware Update

## 4.5. Additional Function

### 4.5.1. AT Command

When pull low CFG(Reload)Pin firstly, T2 will set parameter by port , AT Command is invalid.

When enter into AT Command Mode, then pull down CFG(Reload), T2 will set parameter by port firstly, AT Command is invalid. User pull up or don't connect CFG(Reload), it enter into AT Command Mode again.

### 4.5.2. Display IP and Data

On the webpage of TCP232-T2, it can display the IP of device and sent/received data byte, and the total data byte of TCP232-T2.

In TCP Mode, it can display the TOP 5 device IP and sent/received data byte. In UDP Mode, it only display sent/received data byte. don't display connection IP.

### 4.5.3. Set Client Number in TCP Server Mode

In TCP Server Mode, The maximum number can be configured by user. TCP Client number is from 1 to 16, default value 4. When the Client link is more than 16, the new link will replace the former link from Link 1

When the Client link is more than 4, send and receive data at same time , the data flow should be within 2.5 KB/s.

### 4.5.4. Defined MAC Address

Mac address can be modified. Factory Mac address is exclusive.

### 4.5.5. Defined DNS Server IP

To resolve server domain name, user should send data by gateway or router, then gateway or router Distributes IP address, it can show IP in the webpage. User can set specific domain name resolution IP, to specific gateway or router to resolve domain name.

### 4.5.6. Defined Registration Package

The content of registration package can be defined, 40 bytes at most.

It includes sending registration package when connecting and carrying registration package when sending data. It can be used singly or together.

It is set by webpage, support decimal input and hexadecimal input. By default ,this function isn't checked.

### 4.5.7. Defined Heartbeat Package

The content of heartbeat package can be defined, 40 bytes at most. Time set from 1s to 255s

It can ensure connection is reliable, put an end to connect feign death; It can send to LAN Port or Serial Port singly or at same time.

## 4.6. Configuration Protocol

Configuration flow: to configure protocol cross segment ,all communication protocol is set by UDP Broadcast, Little-endian .In UDP communication,target port number should be 1500. Local port number is random. All communication protocol is UDP broadcast.

### 4.6.1. Network setting protocol command

Chart 3.7-1 - Query command

Function	Head of data package	Length (command-1 bite)	Command	MAC address ( 6 bite)	User name and (12 bite)	Parameter	Check bit (sum)
Search	FF	01	01	-	-	-	02
Restart	FF	xx	02	[MAC]	[username] [password]	-	xx

Read configuration	FF	xx	03	[MAC]	[username] [password]	-	xx
Basic setting	FF	xx	05	[MAC]	[username] [password]	Basic parameter	xx
Port 0 setting	FF	xx	06	[MAC]	[username] [password]	Port parameter	xx
Factory Default	FF	xx	0b	[MAC]	[username] [password]	-	xx
Cloud setting	FF	xx	0c	[MAC]	[username] [password]	Port parameter	xx
Heartbeat package Setting	FF	xx	21	[MAC]	[username] [password]	Port parameter	xx
Extension setting	-	-	-	-	-	-	-

To confirm command accuracy ,we set algorithm of sending data and check method: sum check . From length byte (include length) to check bit (not include check ) ,the sum is check value, only keep lower byte.

**Search command** : Fixed command FF 01 01 02, sum check 02=01+01

**Restart command** : send (22 byte) FF 13 02 00 71 77 7C 42 2F 61 64 6D 69 6E 00 61 64 6D 69 6E 00 FC,

sum check: FC=13+02.....6E+00,

From 4th to 9th is TCP232-T2 MAC address. The last 12 byte before check bit is user name and password, both is 6 byte, if not, write 0 (the last byte of user name and password should be 0 , the following name and password is the same rule)

**Read Configuration command** :send (22 byte) FF 13 03 00 71 77 7C 42 2F 61 64 6D 69 6E 00 61 64 6D 69 6E 00 FD,

FD= 13 + 03 + 00 + ... + 6E + 00

From 4th to 9th is TCP232-T2 MAC address. The last 12 byte before check bit is user name and password, both is 6 byte, if not, write 0.

**Basic setting command**: it includes 67 basic parameter.

E.g : send FF 56 05 00 71 77 7c 42 2F 61 64 6d 69 6e 00 61 64 6d 69 6e 00 95 63 03 00 00 00 50 00 00 07 00 a8 c0 c9 00 a8 c0 00 FF FF FF 55 6c 6c 2d 66 32 00 00 00 00 00 00 00 00 00 00 61 64 6d 69 6e 00 61 64 6d 69 6e 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04, check byte uses sum check method 04 = 56 + 05 + 00+ ... + 01;

From 4th to 9th is TCP232-T2 MAC address,10th to 22th is user name and password .The following byte is basic setting parameter,the last byte is sum check byte.

**Chart 3.7 -2 - basic setting command**

Name	Byte	Example	Instruction
ucSequenceNum	1	00	Reserve package head
ucCRC	1	00	Reserve package head

ucVersion	1	00	Reserve package head
ucFlags	1	80	8th 0:DHCP; 1:Static IP 7 th 0:Turn off factory default function 1:Turn on factory default function
usLocationURLPort	2	00 00	Not enabled, reserved protocol
usHTTPServerPort	2	50 00	HTTP service port
ucUserFlag	1	00	Not enabled, reserved protocol
ulStaticIP	4	07 00 A8 C0	Static IP
ulGatewayIP	4	C9 00 A8 C0	Gateway
ulSubnetMask	4	00 FF FF FF	Subnet mask
ucModName	14	55 53 52 2D 4B 32 00 00 00 00 00 00 00 00	Module name
Protocol reserved	2	00 00	Must be 0
username	6	61 64 6D 69 6E 00	User name
password	6	61 64 6D 69 6E 00	Password
ucNetSendTime	1	00	Not enabled, reserved protocol
uild	2	01 00	Device ID
uclidType	1	A4	bit0(1): send ID when connecting ; bit1(2): send ID when sending data; bit2(4): RS485; bit4(16): Reset; bit5(32):Link-state; bit6(64): index function; bit7(128):Similar RFC2217 function,
mac_addr	6	00 00 00 00 00 00	Device Mac address (only support query)
DNSGatewayIP	4	01 00 A8 C0	DNS address (only support query)
ucReserved	4	00 00 00 00	Unused

Port configuration: The parameter which is used to configure Lan port and Serial port is 63 byte .

E.g.: send ( 63 byte ) FF 52 06 00 71 77 7C 42 2F 61 64 6D 69 6E 00 61 64 6D 69 6E 00 00 C2 01 00 08 01 01 01 00 00 00 00 8C 4E 2A 20 31 39 32 2E 31 36 38 2E 30 2E 32 30 31 00 00 00 00 00 00 00 00 00 00 00 c9 00 A8C0 00 01 00 00 00 00 00 00 00 05 00 00 00 00 F0,

Check byte method : F0=52 + 06 + ... + 00 ;

From 4th to 9th is TCP232-T2 MAC address,10th to 22th is user name and password.the following is port parameter +1 byte and check bit .

**Chart 3.7-3 - Port setting command**

Name	Byte	Example	Instruction
ulBaudRate	4	00 C2 01 00	Serial port baud rate
ucDataSize	1	08	Serial port data bite (0X05/0x06/0x07/0x08)
ucParity	1	01	Serial port check bite 1:no, 2:odd, 3:even, 4:mark, 5:space





Name	Byte	Example	Instruction
H_R_ucFlags	1	00	1th is 1 : heartbeat package to Lan port. 2th is 1: heartbeat package to Serial port. 3th is 1: send registration package. 4th is 1: data carry registration package. 5th is 1: httpd remove package head. 6th is 1: turn on serial port setting parameter function. 7th is 1: input heartbeat package 16 binary system. 8th is : input registration package 16 binary system.
heart_times	1	3c	heartbeat time
heart_len	1	00	heartbeat package length
register_len	1	00	registration package length
heartbeat	40	00 00	heartbeat package content
register_s	40	00 00	registration package content

**4.6.2. Network echo command.**

Search command return result : (36 byte) FF 24 01 00 00 c0 a8 00 07 00 71 77 7c 42 2F 01 0c 00 00 55 53 52 2d 4b 32 00 00 00 00 00 00 00 00 00 00 F2

Initial value is 0X00, subtract each byte in sequence.

F2= 00 - FF - 24 - 01 - 00 - 4B - ... - 31 - 00 - 00.

**Chart 3.7-5 - Search instruction reply command**

Name	Byte	Example	Instruction
TAG_STATUS	0	FF	Fixed number
Packet_length	1	24	Fixed number



If correct, it return FF 01 06 4B If password is correct 4B = 'K'  
 FF 01 06 50 If password is wrong 50 = 'P'

**Return result of heartbeat registration parameter setting :**

If correct, it return FF 01 06 4B If password is correct 4B = 'K'  
 FF 01 06 50 If password is wrong 50 = 'P'

**Other return result:**

If check sum is wrong , it return "E" + right check sum.

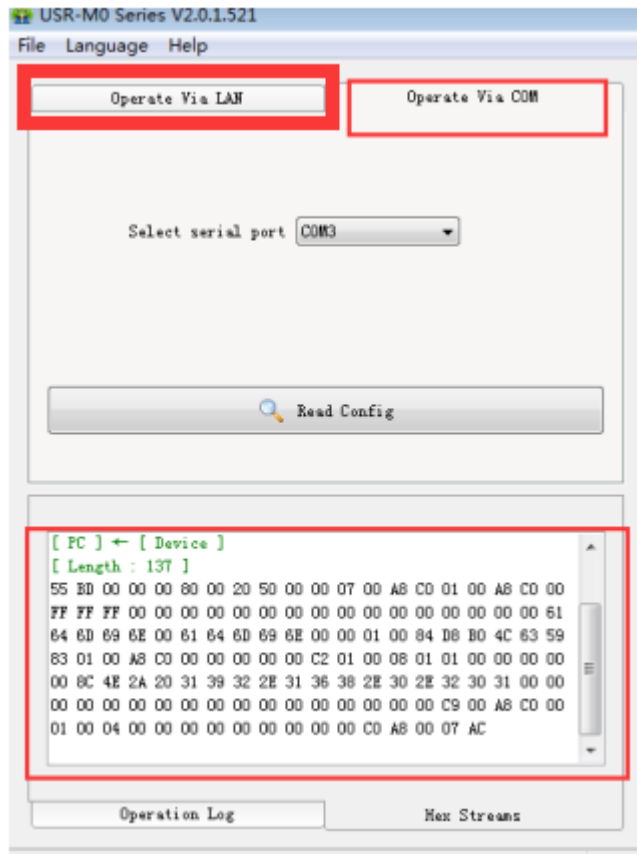
Right operation: FF 01 CMD 'K '

If user name and password is wrong , it return FF 01 CMD 'P'

Other is wrong ,it returns FF 01 CMD'E'

**4.6.3. Monitor function**

If user want to monitor message, please do as follows with USR-M0 software.



**4.6.4. Serial port setting protocol**

Firstly ,user connect Reload(CFG) to the ground, user can refer to setting command or use software ,sending data baud is 9600, 8 data bit, 1 stop bit, (9600,n,8,1) .

No matter what the baud rate ever is , baud rate switches to 9600 idiomatically, send 'U' to port , it means In the status of configuration, after receiving complete data package and check is right ,it returns "K" , if check is wrong, it returns "E" and calculated check bit, this bit is important when sending data by manual test. For Other error,it just returns 'E'.

Restore factory setting Reload(CFG) and port setting Reload(CFG) is the same key. If user want to

restore factory setting, firstly short circuit Reload(CFG) , and port can't send data until 5 seconds later.  
 <Note> after setting ,please pull up Reload(CFG) or don't connect it ,then module can work again.

Data package head for port setting protocol has 3 types:

- ① 55BE write port0 configuration.
- ② 55 BD read T2 configuration.
- ③ 55 BE write basic configuration.
- ④ 55 BB write extended function parameter configuration.
- ⑤ 55 B1 5A restart command;

**Basic configuration** : It includes some serial port parameter configuration. Total 67 byte, check bit xx= 00 + 00 +00+80 ... + 00. Send 55 BE+basic parameter+check bit, If configuration is right ,it returns BE 4B, otherwise ,it returns BE 45.

**Chart 3.7-6 - Port setting command**

Name	Byte	Example	Instruction
ucSequenceNum	1	00	Reserve package head
ucCRC	1	00	Reserve package head
ucVersion	1	00	Reserve package head
ucFlags	1	80	8th 0:DHCP; 1:Static IP 7 th 0:Turn off factory default function 1:Turn on factory default function
usLocationURLPort	2	00 00	Not enabled, reserved protocol
usHTTPServerPort	2	50 00	HTTP service port
ucUserFlag	1	00	Not enabled, reserved protocol
ulStaticIP	4	07 00 A8 C0	Static IP
ulGatewayIP	4	C9 00 A8 C0	Gateway
ulSubnetMask	4	00 FF FF FF	Subnet mask
ucModName	14	55 53 52 2D 4B 32 00 00 00 00 00 00 00 00	Module make
Protocol reserved	2	00 00	Must be 0
username	6	61 64 6D 69 6E 00	User name
password	6	61 64 6D 69 6E 00	Password
ucNetSendTime	1	00	Not enabled, reserved protocol
uild	2	01 00	Device ID
uclidType	1	A4	bit0(1): send ID when connecting ; bit1(2): send ID when sending data; bit2(4): RS485; bit4(16): Reset; bit5(32):Link-state; bit6(64): index function; bit7(128):Similar RFC2217 function,

mac_addrs	6	00 00 00 00 00 00	Device Mac address (only support query)
DNSGatewayIP	4	01 00 A8 C0	DNS address (only support query)
ucReserved	4	00 00 00 00	Unused

Port setting command: it include 65 byte, sum check method . Send 55 BF+basic parameter + check bit. If correct ,it returns BE 4B. Otherwise, it returns BE 45.

**Chart 3.7-7 - Port setting command**

Name	Byte	Example	Instruction
uiBaudRate	4	00 C2 01 00	Serial port baud rate
ucDataSize	1	08	Serial port data bite (0X05/0x06/0x07/0x08)
ucParity	1	01	Serial port check bite 1:no, 2:odd, 3:even, 4:mark, 5:space
ucStopBits	1	01	Serial port stop bite (0x01/0x02)
ucFlowControl	1	00	Not enabled, reserved protocol
uiTelnetTimeout	4	00 00 00 00	Not enabled, reserved protocol
usTelnetLocalPort	2	8C 4E	Local port
usTelnetRemotePort	2	2a 20	Remote port
uiTelnetURL	30	31 39 32 2E 31 36 38 2E 30 2E 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	IP address or domain name send by ASCII E.g.: 192.168.0.1
uiTelnetIPAddr	4	00 00 00 00	Invalid byte , arbitrary configuration
ucFlags	1	00	Cloud function: For 5th , 0: turn off cloud      1: turn on cloud
ucWorkMode	1	01	Work mode : 0: UDP, 1:TCP Client, 2: UDP Server, 3: TCP Server 4: HTTPD Client
HTPucFlags	1	00	0:HTTPD GET;    1:HTTPD POST
tc_number	1	04	Client number that TCP Server can connect
uiPackLen	2	00 00	Not enabled, reserved protocol
ucPackTime	1	0A	Serial port packing time
ucTimeCount	1	91	Write the read data as it is .
ucReserved	5	00 00 00 00 00	Write 00 00 00 00 00. For extension use

Read configuration command : TCP232-T2 send 55 BD, return value is 137 byte, there is no check bit. TCP232-T2 return parameter is as follows.

**Chart 3.7- 8 - Read configuration command**

Name	Byte	Example	Instruction
ucSequenceNum	2	55 bd	Reserve package head
ucCRC	1	00	Reserve package head
ucVersion	1	00	Reserve package head

ucFlags	1	80	8th 0:DHCP; 1:Static IP 7th 0:Turn off factory default function 1:Turn on factory default function
usLocationURLPort	2	00 00	Not enabled, reserved protocol
usHTTPServerPort	2	50 00	HTTP service port
ucUserFlag	1	00	Not enabled, reserved protocol
ulStaticIP	4	07 00 A8 C0	Static IP
ulGatewayIP	4	C9 00 A8 C0	Gateway
ulSubnetMask	4	00 FF FF FF	Subnet mask
ucModName	14	55 53 52 2D 4B 32 00 00 00 00 00 00 00 00	Module name
Protocol reserved	2	00 00	Must be 0
username	6	61 64 6D 69 6E 00	User name
password	6	61 64 6D 69 6E 00	Password
ucNetSendTime	1	00	Not enabled, reserved protocol
uuld	2	01 00	Device ID
uuldType	1	A4	bit0(1): send ID when connecting ; bit1(2): send ID when sending data; bit2(4): RS485; bit4(16): Reset; bit5(32):Link-state; bit6(64): index function; bit7(128):Similar RFC2217 function,
mac_addr	6	00 00 00 00 00 00	Device Mac address (only support query)
DNS Gateway IP	4	01 00 A8 C0	DNS address (only support query)
ucReserved	4	arbitrary value	Unused
ulBaudRate	4	00 C2 01 00	Serial port baud rate
ucDataSize	1	08	Serial port data bite (0X05/0x06/0x07/0x08)
ucParity	1	01	Serial port check bite 1:no, 2:odd, 3:even, 4:mark, 5:space
ucStopBits	1	01	Serial port stop bite (0x01/0x02)
ucFlowControl	1	00	Not enabled, reserved protocol
ulTelnetTimeout	4	00 00 00 00	Not enabled, reserved protocol
usTelnetLocalPort	2	8C 4E	Local port
usTelnetRemotePort	2	2a 20	Remote port
uiTelnetURL	30	31 39 32 2E 31 36 38 2E 30 2E 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	IP address or domain name send by ASCII E.g.: 192.168.0.1

uTelnetIPAddr	4	00 00 00 00	Not enabled,
ucFlags	1	00	Cloud function: For 5th , 0: turn off cloud      1: turn on cloud
ucWorkMode	1	01	Work mode : 0: UDP, 1:TCP Client, 2: UDP Server, 3: TCP Server 4: HTTPD Client
HTPucFlags	1	00	0: HTTPD GET;    1:HTTPD POST
tc_number	1	04	Client number that TCP Server can connect
uiPackLen	2	00 00	Not enabled, reserved protocol
ucPackTime	1	0A	Serial port packing time
ucTimeCount	1	91	Write the read data as it is .
ucReserved	5	00 00 00 00 00	Write 00 00 00 00 00. For extension use
Current IP	4	07 00 A8 C0	Current module IP address
Version	1	01	Version number

Serial port configuration extended function parameter: extended function 112 character, send 55 BB + special function character + check bit. If correct , it returns BB 4B. Otherwise It returns BB 45.

**Chart 3.7-9 - Serial port setting extended function parameter command**

Name	Byte	Example	Instruction
Device ID	20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Module Cloud ID
CLOUD_password	8	00 00 00 00 00 00 00 00	Module Cloud password
H_R_ucFlags	1	00	1th is 1 : heartbeat package to Lan port. 2th is 1: heartbeat package to Serial port. 3th is 1: send registration package. 4th is 1: data carry registration package. 5th is 1: httpd remove package head. 6th is 1: turn on serial port setting parameter function. 7th is 1: input heartbeat package 16 binary system. 8th is : input registration package 16 binary system.
heart_times	1	3c	heartbeat time
heart_len	1	00	heartbeat package length
register_len	1	00	registration package length
heartbeat	40	00 00	heartbeat package content

		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
register_s	40	00 00	registration package content

#### 4.6.5. Reset command

Send 55 B1 5A, if correct , it returns BA 4B, then module restart. If wrong, it won't return.

#### 4.6.6. Similar T24 series setting protocol

Network command : Fixed format 40 byte data package, please refer to the following format and data , TCP232-T2 will return 35 bytes data package. 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39. When setting parameter ,it send 40 bytes, returns 35 bytes.

**Chart 3.7-10 -Network command**

Name	Byte	Example	Instruction
MAC	6	00 CE 83 25 4D 60	Module MAC that need to be set
Old password	6	31 31 30 34 31 35	Password for network module, 110415 is initial password.
Target IP	4	C9 00 A8 C0	Target IP
Target Port	2	2A 20	Target Port
Module IP	4	07 00 A8 C0	Module IP
Module Port	2	8C 4E	Module Port
Gateway	4	C9 00 A8 C0	Gateway IP
Work mode	1	01	0:UDP Client, 1:TCP Client 2:UDP Server, 3:TCP Server
Baud rate	3	00 C2 01	Port baud rate
Port parameter bit	1	03	Data bit, stop bit, check bit
Independent ID	3	00 00 00	ID-H,ID-L,ID-type, if don't need, write 0 ( ID type has addition meaning ,refer to appendix I )
Subnet mask	4	00 FF FF FF	Subnet mask, Little Endian

**Chart 3.7- 11- Network return parameter**

Name	Byte	Example	Instruction
MAC	6	00 CE 83 25 4D 60	Module MAC that need to be set
Version	1	42	Version



Target IP	4	C9 00 A8 C0	Target IP
Target Port	2	2A 20	Target Port
Module IP	4	07 00 A8 C0	Module IP
Module Port	2	8C 4E	Module Port
Gateway	4	C9 00 A8 C0	Gateway IP
Work mode	1	01	0:UDP Client, 1:TCP Client 2:UDP Server, 3:TCP Server
Baud rate	3	00 C2 01	Port baud rate
Port parameter bit	1	03	Data bit, stop bit, check bit
Independent ID	3	00 00 00	ID-H,ID-L,ID-type, if don't need, write 0 ( ID type has addition meaning ,refer to appendix I )
Subnet mask	4	00 FF FF FF	Subnet mask, Little Endian

Port parameter set : Port parameter set of similar T24 set protocol.

Read parameter command: send 55BC to TCP232, return message and set

Parameter command as follows:

**Chart 3.7-12 -Read parameter return command**

Name	Byte	Example	Instruction
Head of data package	2	55 BC	Head of data package
Target IP	4	C9 00 A8 C0	Target IP
Target Port	2	2A 20	Target Port
Module IP	4	07 00 A8 C0	Module IP
Module Port	2	8C 4E	Module Port
Gateway	4	C9 00 A8 C0	Gateway IP
Work mode	1	01	0:UDP Client, 1:TCP Client 2:UDP Server, 3:TCP Server
Baud rate	3	00 C2 01	Port baud rate
Port parameter bit	1	03	Data bit, stop bit, check bit
Independent ID	3	00 00 00	ID-H,ID-L,ID-type, if don't need, write 0 ( ID type has addition meaning ,refer to appendix I )
Subnet mask	4	00 FF FF FF	Subnet mask, Little Endian
Firmware version	1	58	Firmware version lowest byte
Sum Check	1	B9	Sum check, from target IP to sum check (result keep low byte )

**Chart 3.7- 13 - Set parameter command**

Name	Byte	Example	Instruction
Head of data package	2	55 BA	Head of data package
Target IP	4	C9 00 A8 C0	Target IP
Target Port	2	2A 20	Target Port
Module IP	4	07 00 A8 C0	Module IP
Module Port	2	8C 4E	Module Port
Gateway	4	C9 00 A8 C0	Gateway IP
Work mode	1	01	0:UDP Client, 1:TCP Client 2:UDP Server, 3:TCP Server
Baud rate	3	00 C2 01	Port baud rate
Port parameter bit	1	03	Data bit, stop bit, check bit
Independent ID	3	00 00 00	ID-H,ID-L,ID-type, if don't need, write 0 ( ID type has addition meaning ,refer to appendix I )
Subnet mask	4	00 FF FF FF	Subnet mask, Little Endian
Sum Check	1	61	Sum check, from target IP to sum check (result keep low byte )

## 4.7. UART Framing mechanism

### 4.7.1. packing method

TCP232-T2 accepts time packing method.

1) packing time standard: more than 4 byte packing time.

2) Calculation method :

√ TCP232-T2 Port parameter :data bit 8 byte, stop bit 1 byte, start bit 1byte.

√ Length of one data : 8+1+1=10

√ Packing time of four byte:

$$T = \frac{1}{\text{Baud rate}} * 10 * 4$$

√ Port packing length : 400 byte.

√ When baud rate is 115200, default packing time 0.4ms.

### 4.7.2. Flow calculation

When TCP232-T2 work in Cloud mode ,it receives network data, then it sends to Port. Because port speed is limited, sometimes data will overflow.

E.g network data will send m byte data every n seconds. To check whether data can overflow : (support network is in good station ,network data transmission time is negligible ) If data won't overflow , m byte data can be finished in n seconds.

M byte data transmission time :

$$T = \frac{1}{\text{Baud rate}} * 10 * m$$

If  $n > 2T$  ,data won't overflow, TCP232-T2 work well.

## 5. Parameter Setting

At first, user should connect PC to TCP232-T2 Lan port ,then set parameter by webpage or software. TCP232-T2 connects Auto-MDI/MDIX RJ45 port with 10/100Mbps. It support Router connection and direct connection.

- 1)TCP232-T2 connect PC by Lan cable, user should set parameter for PC. When TCP232-T2 connects to power supply and PC, module will connect PC directly. (Module IP and PC IP should in the same Network segment )
- 2)TCP232-T2 connects PC by router, one PC can connect several modules or one module connects Several PC (When modules works as TCP Server , at most it can connect 16 clients )

### 5.1. Webpage Setting

#### 5.1.1. Log in

Open a browser, type 192.168.0.7 , Name and password: admin

User can also log in by software.

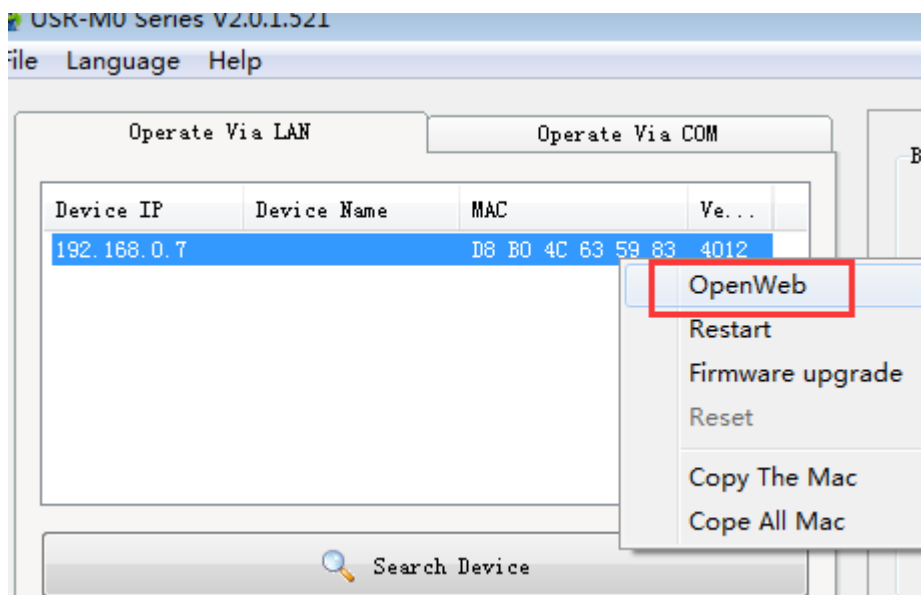


Diagram 5.1.1-1 Open Web

### 5.1.2. State Configuration

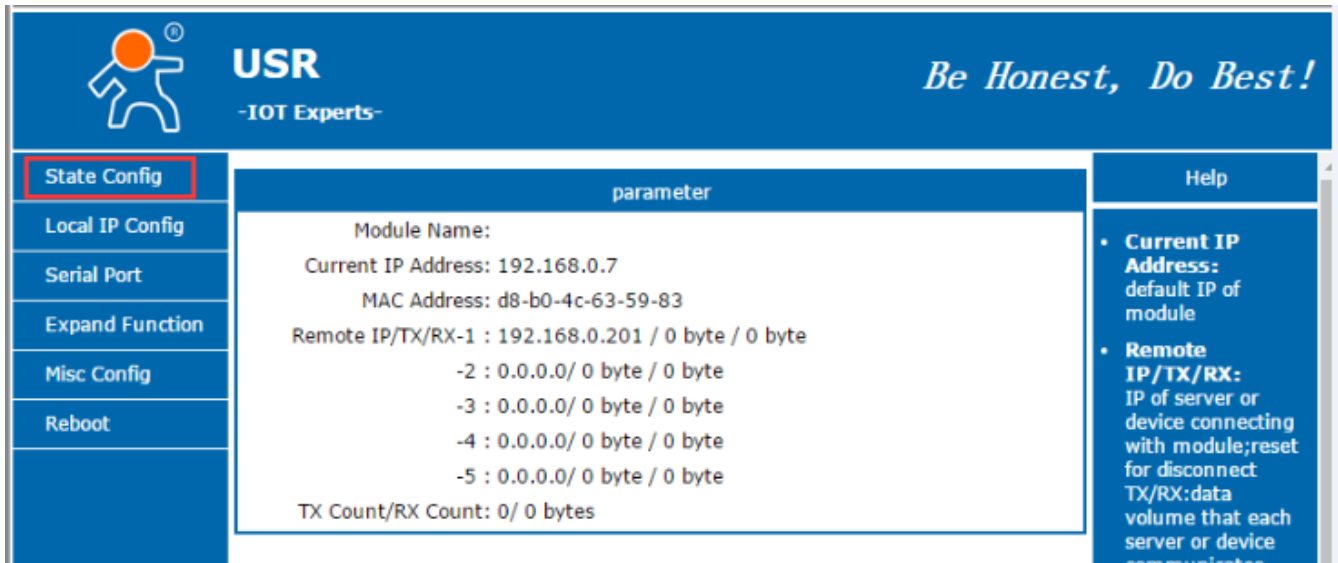


Diagram 5.1.2-1 State Configuration

### 5.1.3. Local IP

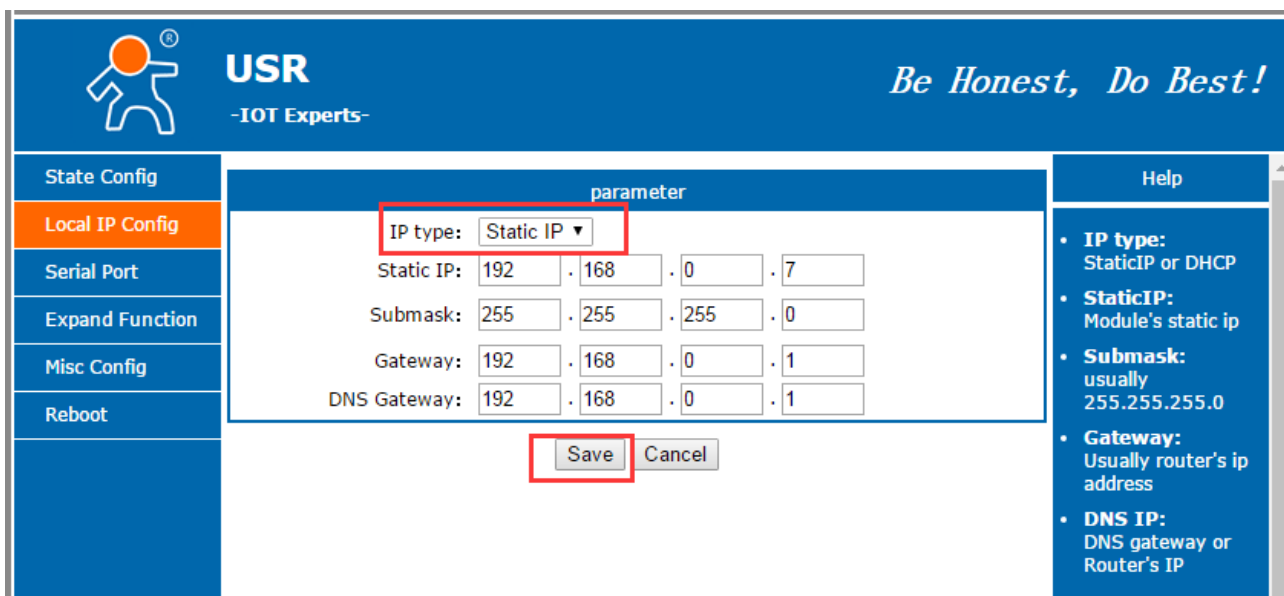


Diagram 5.1.3-1 Local IP

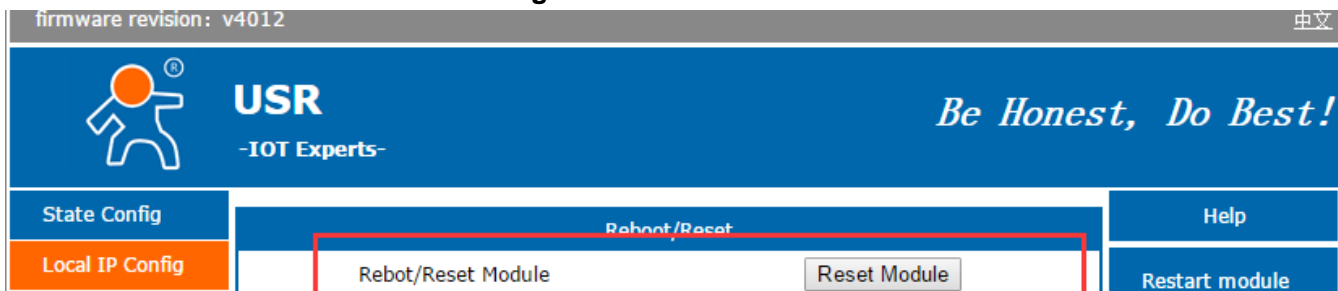


Diagram 5.1.3-2 Local IP

### 5.1.4. Serial Port

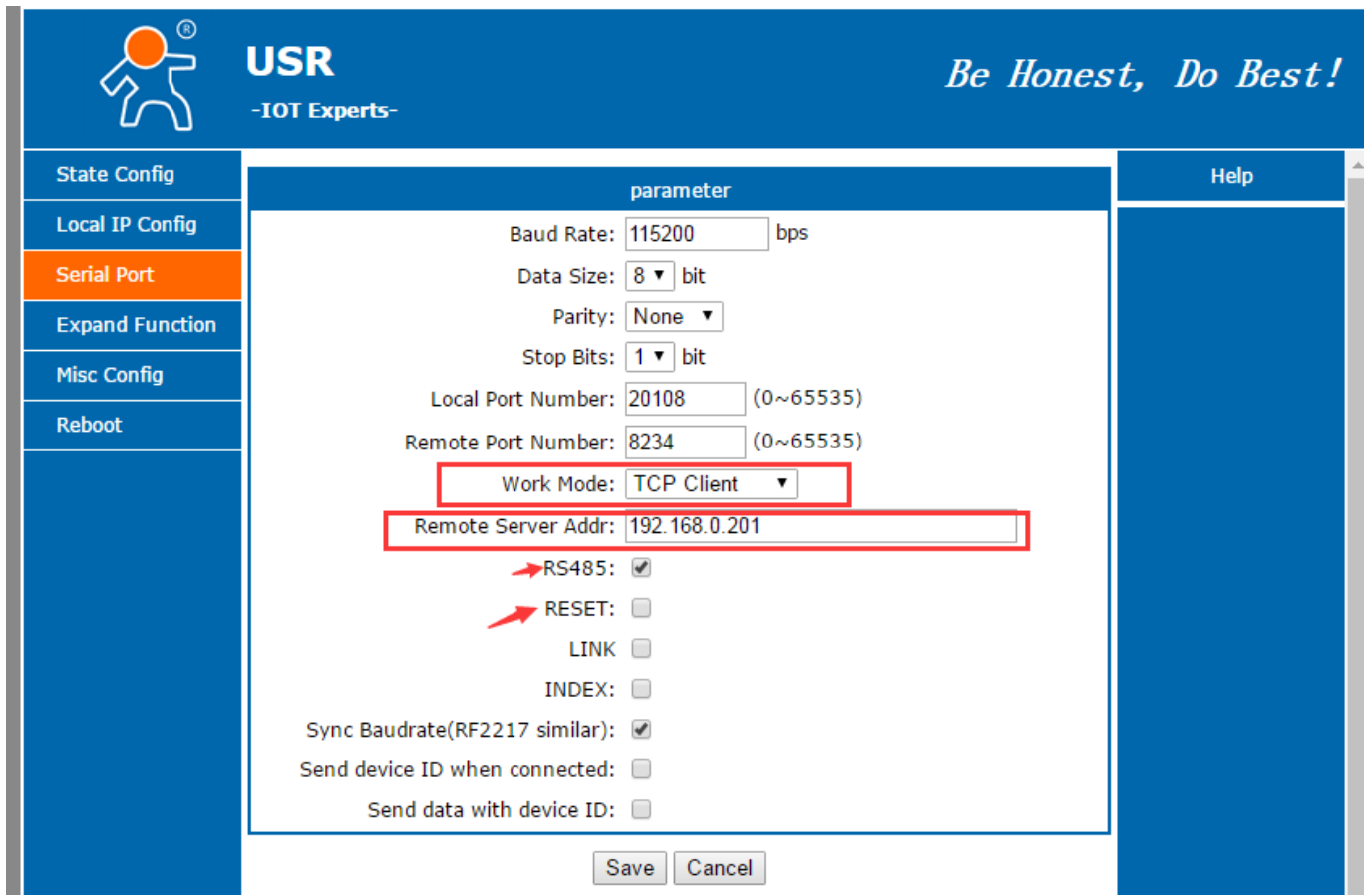


Diagram 5.1.4-1 Serial Port

### 5.1.5. Expand Function

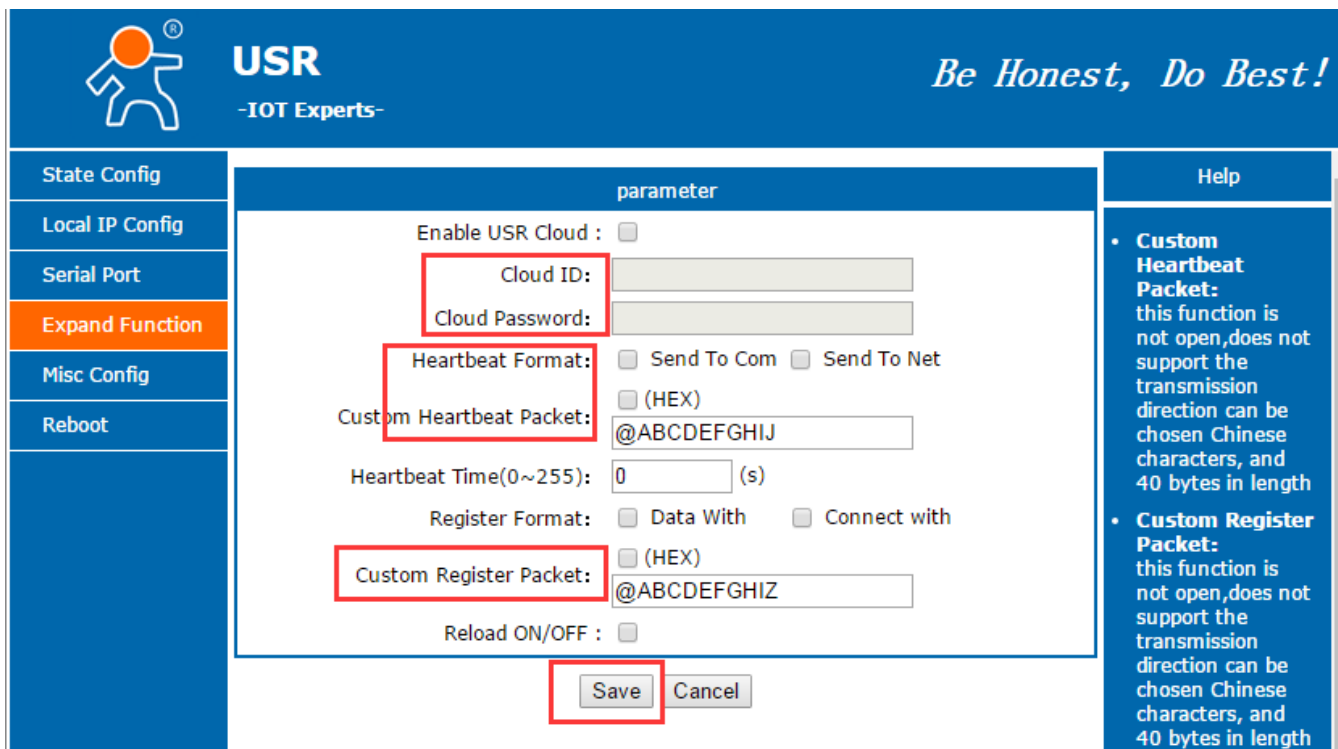


Diagram 5.1.5-1 Expand Function

### 5.1.6. Misc Configuration

Set module name, user name, pass word, MAC.

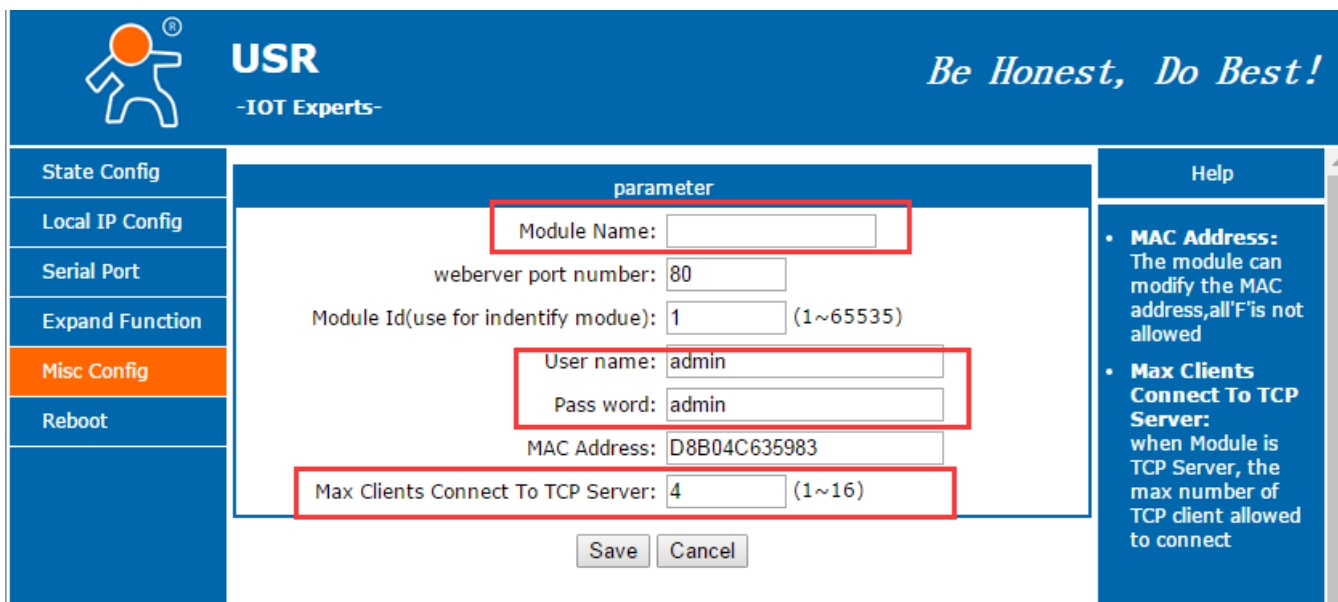


Diagram 5.1.6-1 Misc Setting

### 5.1.7. Reboot

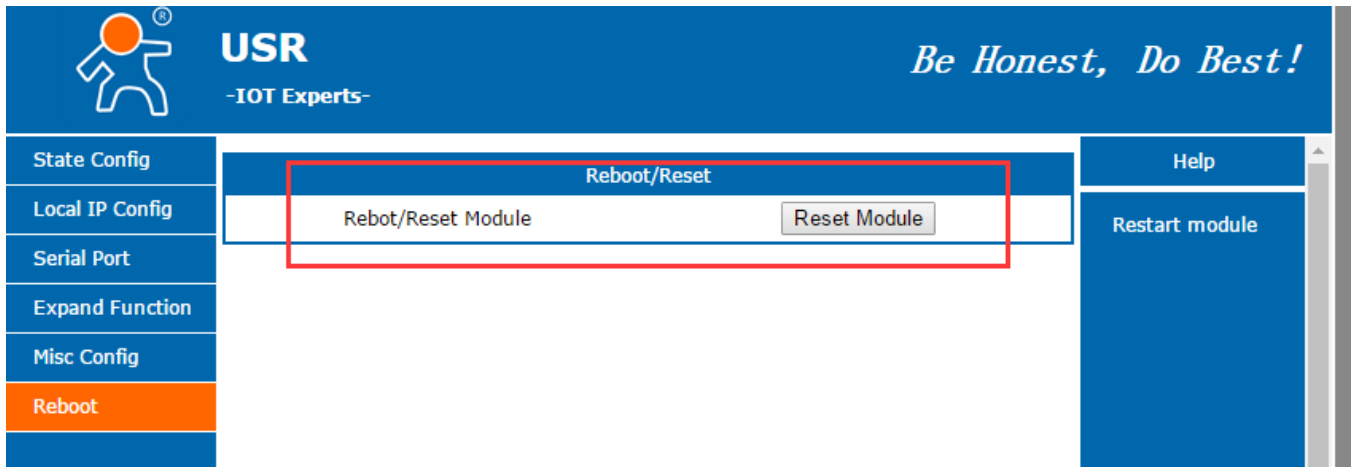


Diagram 5.1.7-1 Reboot Setting

## 5.2. Software Setting

- 1) Search device

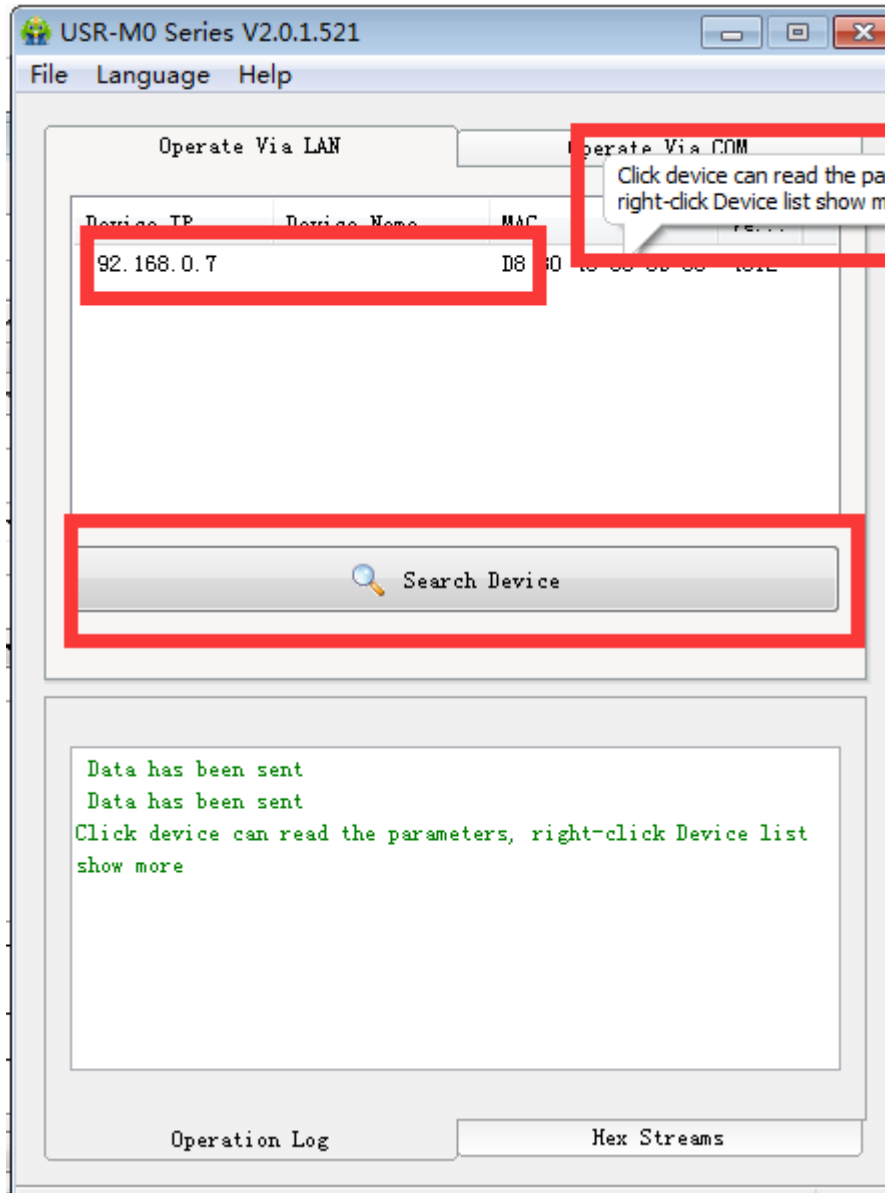


Diagram 5.2-1 Search

2) Set parameters



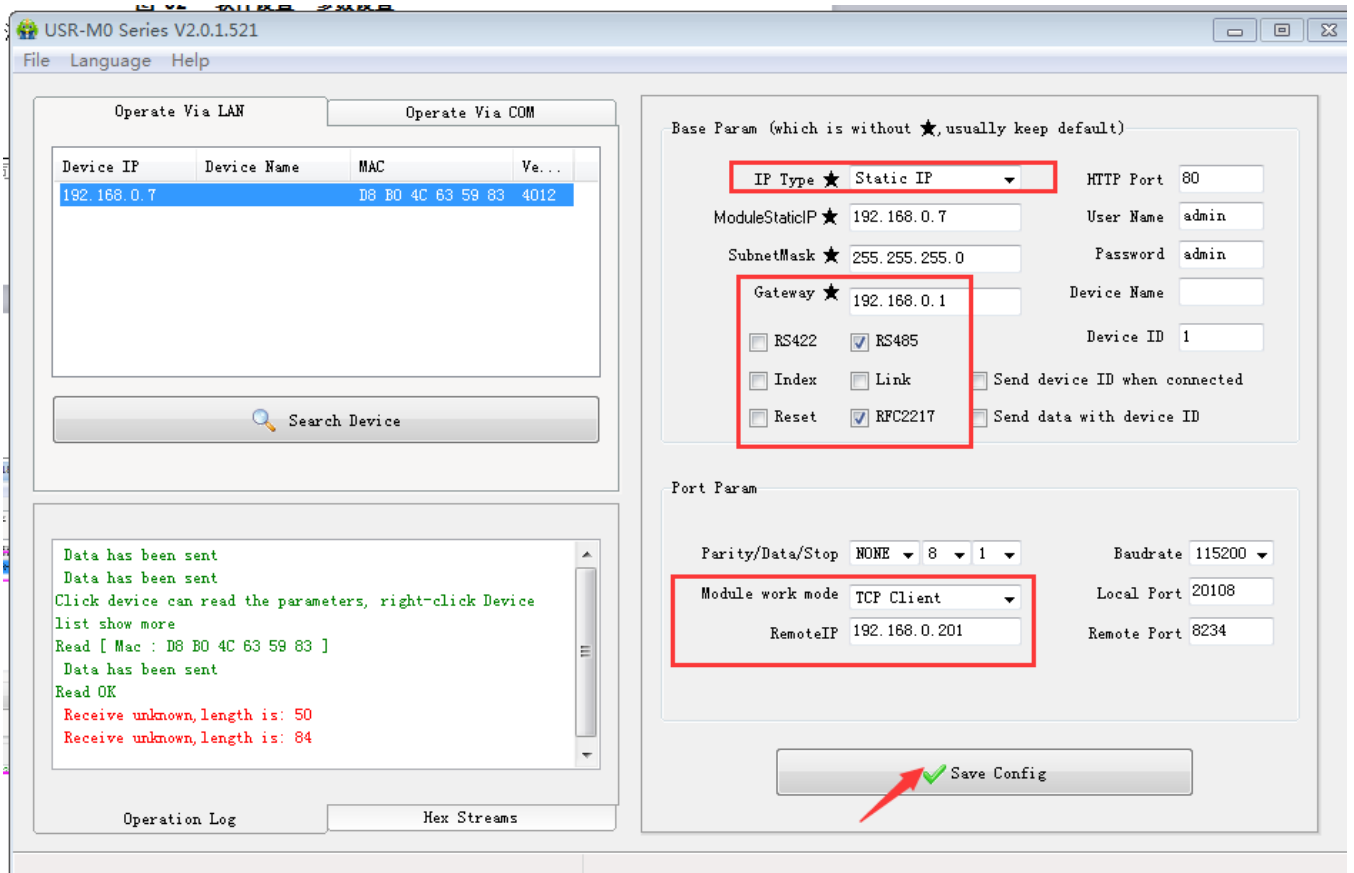


Diagram 5.2-2 Parameter setting

3) Check data

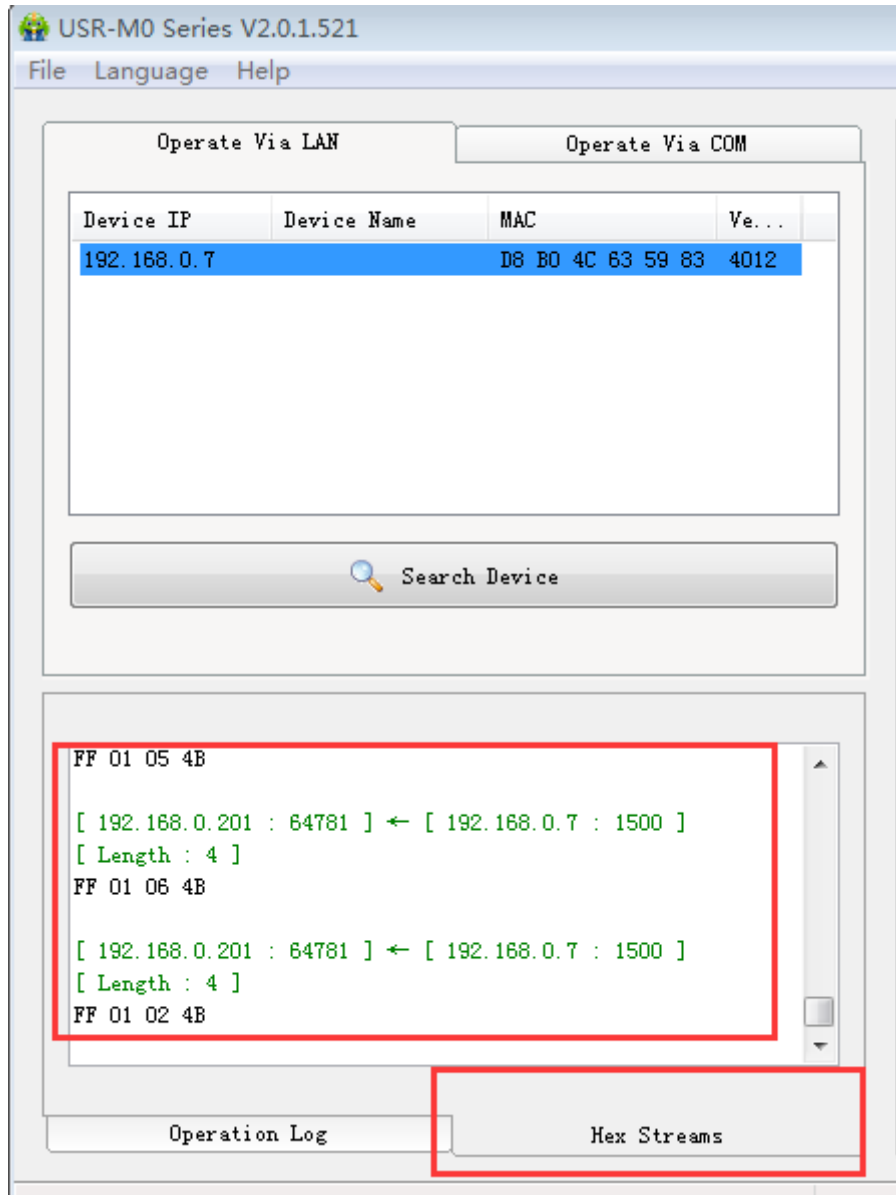


Diagram 5.2-3 Check Streams

4) Press CFG (Reload) and click “operate via COM”

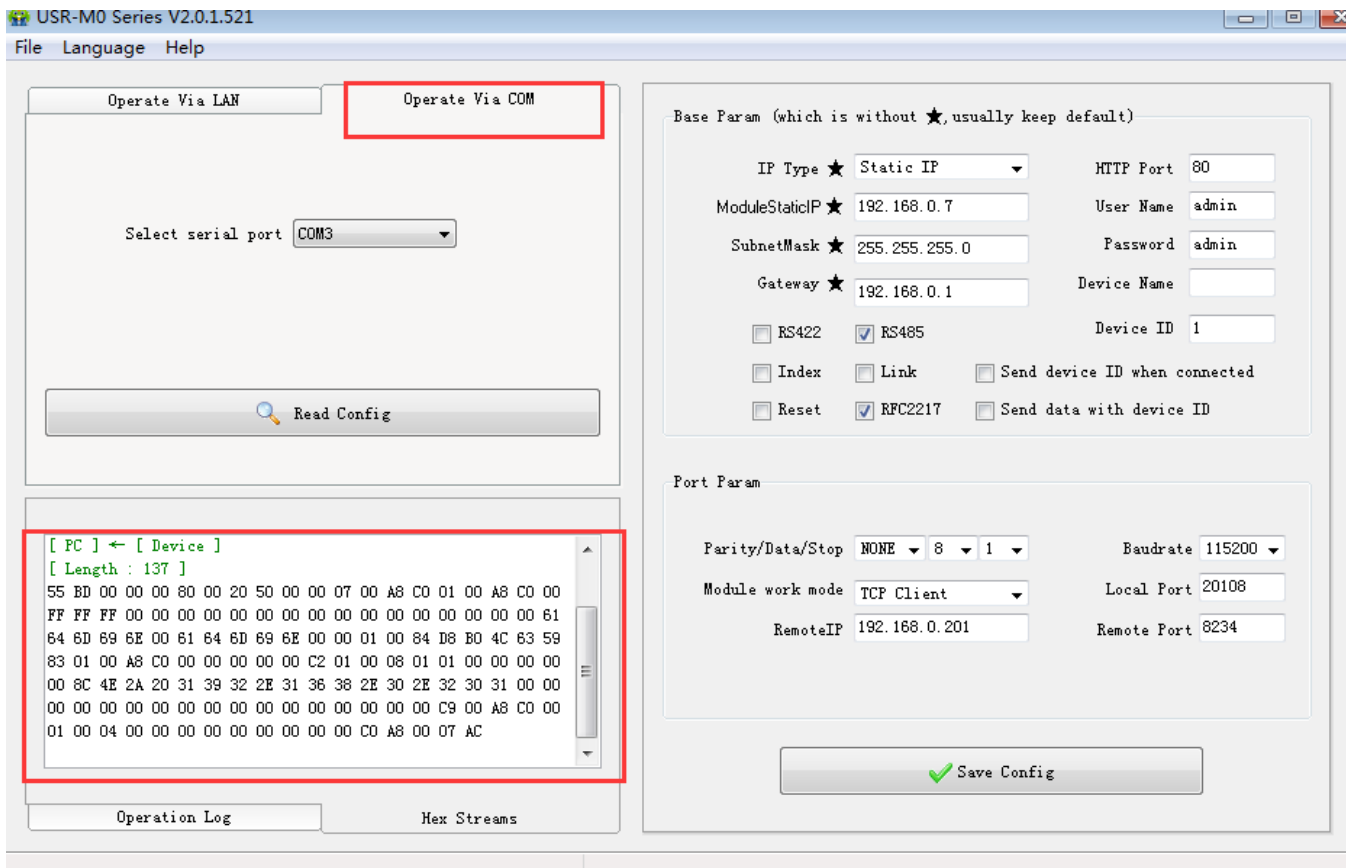


Diagram 5.2-4 Port Setting

### 5.3. AT Command

After entering into AT Command , Cloud function is forbidden until quits the mode.

When pull down CFG(Reload), T2 will do read setting through port to parameter, AT command won't work. If user has pull down CFG(Reload) to AT mode, serial configuration is preferred , AT command won't work, when pull up CFG(Reload) or don't connect it, it enter into AT mode again.

#### 5.3.1. AT Command Model

- 1) Send +++ to T2 module from serial port, T2 send "a "
- 2) Send "a" in 3s after receiving "a".
- 3) T2 return to + OK , enter into AT Demand Model.

#### 5.3.2. AT Command Set

##### 5.3.2.1. AT+E

Function: Query / Set AT Command echo setting

Format: Query : A+E<CR>  
 <CR><LF>+OK=<on/off><CR><LF>  
 Set: A+E=<on/off><CR>

<CR><LF>+OK=<on/off><CR><LF>

Parameter: 1.on : turn on echo function, echo command in AT command.  
2.off : turn off echo function.

### 5.3.2.2. AT +ENTM

Function: enter into transparent transmission

Format:

Set  
AT+ENTM<CR>  
<CR><LF>+OK<CR><LF>

Parameter : Null ( when execute this commend, module switch into Cloud mode )

### 5.3.2.3. AT+Z

Function: restart module

Format :

Set  
AT+Z<CR>  
<CR><LF>+OK<CR><LF>

Parameter : Null ( when execute this commend, module restarts ,quit AT Command )

### 5.3.2.4. AT+VER

Function: Set/Query module firmware version

Format: Query : AT+VER<CR>

<CR><LF>+OK=<ver><CR><LF>

Parameter: ver: query module

### 5.3.2.5. AT+MAC

Function: query module MAC

Format:

Query  
AT+MAC<CR>  
<CR><LF>+OK=<MAC><CR><LF>

Parameter : mac module MAC Address

### 5.3.2.6. AT+CLEAR

Function: factory default

Format:

Set

AT+CLEAR<CR>

<CR><LF>+OK<CR><LF>

### 5.3.2.7. AT+UART

Function: set/query UART parameter.

Format:

Query: AT+UART<CR>

<CR><LF>+OK=<baudrate,data\_bits,stop\_bit,parity,flowctrl, uartnum><CR><LF>

Set: AT+UART=<baudrate,data\_bits,stop\_bit,parity,flowctrl, uartnum ><CR><LF>

<CR><LF>+OK<CR><LF>

Parameter: baud rate: 9600,19200,38400,57600,115200,230400,380400,460800(module can set baud rate randomly )

Data bits :5,6,7,8

Stop bits: 1,1.5,2 ( refer to module )

Parity: check bits

None( no check bits )

Even( even check)

ODD(odd check )

Mask (1 check )

Space (0 check )

Flow control : hardware flow control ( CTS RTS) ,T2 has no hardware flow control, can only set NFC

NFC: no hardware flow control .

FC: has hardware flow control .

485: support 485 , when it turns on , RS\_EN pin is the same as RTS pin.

UART number : the port number need to set ( T2 only has 1 port, so write 0 here )

### 5.3.2.8. AT SOCKA1

Function: set/query UART parameter.

Format:

Query: AT+UART<CR>

<CR><LF>+OK=<baudrate,data\_bits,stop\_bit,parity,flowctrl, uartnum><CR><LF>

Set: AT+UART=<baudrate,data\_bits,stop\_bit,parity,flowctrl, uartnum ><CR><LF>

<CR><LF>+OK<CR><LF>

Parameter: protocol: protocol type ,includes:

TCPS --TCP Server

TCPC-- TCP Client

UDPS-- UDP Server

UDPC-- UDP Client

IP: when the module is set “client” , server IP address.  
 Port :protocol port ,decimal number ,less than 65535.

**5.3.2.9. AT+DHCPEN**

Function: turn on /off DHCP

Format:

```
Query: AT+DHCPEN<CR>
      <CR><LF>+OK=<Type><CR><LF>
Set:AT+DHCPGW=<Type><CR>
     <CR><LF>+OK<CR><LF>
```

Parameter : Type: on means turn on DHCP , off means turn off DHCP.

**5.3.2.10. AT+WANN**

Function: set/query WAN IP

Format:

```
Query:
AT+WANN<CR>
<CR><LF>+OK=<mode,address,mask,gateway><CR><LF>
Set:
AT+WANN=<mode,address,mask,gateway><CR>
<CR><LF>+OK<CR><LF>
```

Parameter:

mode:Network IP mode( T2 can only set static mode)

Static: static IP

DHCP: dynamic IP (address,mask,gateway can be omitted)

Address : IP address

Mask: subnet mask

Gate way: gateway address

**6. Appendix**

**Appendix I :Meaning of Port parameter bit**

Bit number	Instruction	Value	Description
1:0	Data bit choice	00	5 bit data bit
		01	6 bit data bit
		10	7 bit data bit
		11	8 bit data bit
2	Stop bit	00	1 bit stop bit
		01	2 bit stop bit

3	Check bit enable	00	Unabled check bit
		01	Enabled check bit
5:4	Check bit type	00	ODD check
		01	EVEN check
		10	Set 1
		11	Zero Clearing
8:6	Undefined	000	Write 0

## Appendix II: ID type of Independent ID

This byte is the last byte of 3 bytes, additional meaning is as follows:

bit0(1) send ID when connecting;

bit1(2) send ID when send data;

bit2(4) RS485;

bit3(8) NC;

bit4(16) Reset;

bit5(32) Link-state;

bit6(64) tcp server index;

bit7(128) baud rate synchronization, similar RFC2217.

When each bit is 1, it means related function is invalid. Otherwise, it is valid, default is RS485 and similar RFC2217 set 1 .(turn on function, value is 0x84 )

## Appendix III: Upper computer Socket programming example

### Server Socket code:

```

1. #include <stdio.h>
2. #include <string.h>
3. #include <sys/socket.h>
4. #include <netinet/in.h>
5. #include <stdlib.h>
6. #include <syslog.h>
7. #include <errno.h>
8. #define MAX_LISTEN_NUM 5
9. #define SEND_BUF_SIZE 100
10. #define RECV_BUF_SIZE 100
11. #define LISTEN_PORT 1010
12. int main()
13. {
14.     int listen_sock = 0;
15.     int app_sock = 0;
16.     struct sockaddr_in hostaddr;
17.     struct sockaddr_in clientaddr;
18.     int socklen = sizeof(clientaddr);

```

```
19. char sendbuf[SEND_BUF_SIZE] = {0};
20. char recvbuf[RECV_BUF_SIZE] = {0};
21. int sendlen = 0;
22. int recvlen = 0;
23. int retlen = 0;
24. int leftlen = 0;
25. char *ptr = NULL;
26. memset((void *)&hostaddr, 0, sizeof(hostaddr));
27. memset((void *)&clientaddr, 0, sizeof(clientaddr));
28. hostaddr.sin_family = AF_INET;
29. hostaddr.sin_port = htons(LISTEN_PORT);
30. hostaddr.sin_addr.s_addr = htonl(INADDR_ANY);
31. listen_sock = socket(AF_INET, SOCK_STREAM, 0);
32. if(listen_sock < 0)
33. {
34.     syslog(LOG_ERR, "%s:%d, create socket failed", __FILE__, __LINE__);
35.     exit(1);
36. }
37. if(bind(listen_sock, (struct sockaddr *)&hostaddr, sizeof(hostaddr)) < 0)
38. {
39.     syslog(LOG_ERR, "%s:%d, bind socket failed", __FILE__, __LINE__);
40.     exit(1);
41. }
42. if(listen(listen_sock, MAX_LISTEN_NUM) < 0)
43. {
44.     syslog(LOG_ERR, "%s:%d, listen failed", __FILE__, __LINE__);
45.     exit(1);
46. }
47. while(1)
48. {
49.     app_sock = accept(listen_sock, (struct sockaddr *)&clientaddr, &socklen);
50.     if(app_sock < 0)
51.     {
52.         syslog(LOG_ERR, "%s:%d, accept failed", __FILE__, __LINE__);
53.         exit(1);
54.     }
55.     sprintf(sendbuf, "welcome %s:%d here!\n", inet_ntoa(clientaddr.sin_addr.s_addr), clientaddr.sin_port);
56.     //send data
57.     sendlen = strlen(sendbuf) + 1;
58.     retlen = 0;
59.     leftlen = sendlen;
60.     ptr = sendbuf;
61.     //while(leftlen)
62.     {
63.         retlen = send(app_sock, ptr, sendlen, 0);
```



```
64.     if(retlen < 0)
65.     {
66.         if(errno == EINTR)
67.             retlen = 0;
68.         else
69.             exit(1);
70.     }
71.     leftlen -= retlen;
72.     ptr += retlen;
73. }
74. //receive data
75. recvlen = 0;
76. retlen = 0;
77. ptr = recvbuf;
78. leftlen = RECV_BUF_SIZE -1;
79. //do
80. {
81.     retlen = recv(app_sock, ptr, leftlen, 0) ;
82.     if(retlen < 0)
83.     {
84.         if(errno == EINTR)
85.             retlen = 0;
86.         else
87.             exit(1);
88.     }
89.     recvlen += retlen;
90.     leftlen -= retlen;
91.     ptr += retlen;
92. }
93. //while(recvlen && leftlen);
94. printf("receive data is : %s", recvbuf);
95. close(app_sock);
96. }
97. close(listen_sock);
98.
99. return 0;
100. }
```

### Client Socket code:

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <sys/socket.h>
4. #include <netinet/in.h>
5. #include <syslog.h>
```

```
6. #include <errno.h>
7. #include <stdlib.h>
8. #define MAX_LISTEN_NUM 5
9. #define SEND_BUF_SIZE 100
10. #define RECV_BUF_SIZE 100
11. #define SERVER_PORT 1010
12. int main()
13. {   int sock_fd = 0;
14.     char recvbuf[RECV_BUF_SIZE] = {0};
15.     char sendbuf[SEND_BUF_SIZE] = {0};
16.     int recvlen = 0;
17.     int retlen = 0;
18.     int sendlen = 0;
19.     int leftlen = 0;
20.     char *ptr = NULL;
21.     struct sockaddr_in ser_addr;
22.     memset(&ser_addr, 0, sizeof(ser_addr));
23.     ser_addr.sin_family = AF_INET;
24.     inet_aton("127.0.0.1", (struct in_addr *)&ser_addr.sin_addr);
25.     ser_addr.sin_port = htons(SERVER_PORT);
26.     sock_fd = socket(AF_INET, SOCK_STREAM, 0);
27.     if(sock_fd < 0)
28.     {
29.         syslog(LOG_ERR, "%s:%d, create socket failed", __FILE__, __LINE__);
30.         exit(1);
31.     }
32.     if(connect(sock_fd, (struct sockaddr *)&ser_addr, sizeof(ser_addr)) < 0)
33.     {
34.         syslog(LOG_ERR, "%s:%d, connect socket failed", __FILE__, __LINE__);
35.         exit(1);
36.     }
37.     //receive data
38.     recvlen = 0;
39.     retlen = 0;
40.     ptr = recvbuf;
41.     leftlen = RECV_BUF_SIZE -1;
42.     //do
43.     {
44.         retlen = recv(sock_fd, ptr, leftlen, 0) ;
45.         if(retlen < 0)
46.         {
47.             if(errno == EINTR)
48.                 retlen = 0;
49.             else
50.                 exit(1);
```

```
51.     }
52.     recvlen += retlen;
53.     leftlen -= retlen;
54.     ptr += retlen;
55.     }
56.     //while(recvlen && leftlen);
57.     printf("receive data is : %s", recvbuf);
58.     sprintf(sendbuf, "hello server/n");
59.     //send data
60.     sendlen = strlen(sendbuf) +1;
61.     retlen = 0;
62.     leftlen = sendlen;
63.     ptr = sendbuf;
64.     // while(leftlen)
65.     {
66.         retlen = send(sock_fd, ptr, sendlen, 0);
67.         if(retlen < 0)
68.         {
69.             if(errno == EINTR)
70.                 retlen = 0;
71.             else
72.                 exit(1);
73.         }
74.         leftlen -= retlen;
75.         ptr += retlen;
76.     }
77.     close(sock_fd);
78. }
79.
```

## 7. Undated History

V 1.0 2016-4-29 First Version