Serial Device Server

User Manual

2 ports RS232/485/422 to TCP/IP converter

Modbus Gateway

P/N: ITT-SRLAN5200

Contents

Summary

ITTLAN5240 serial device server is a 2 ports protocol converter between RS232/485/422 and TCP/IP developed by Shanghai ITTLAN. ITTLAN5240 support 2 RS232 ports, 2 RS422/485 ports, can realize 2 serial ports simultaneous full-duplex working through connecting to ITTLAN5240 via a cable. The RS485 interface is green terminal interface, the RS232/RS485 are RJ45 Ethernet port connection interface. ITTLAN can equip RJ45 to DB9 cable, 2 such line connecting to 2 RJ45 RS232 interface can lead out 2 DB9 male RS232 splice. Also 5240 provide an extra Ethernet interface, can be used as switch or cascade. ITTLAN5240 support extending to 4/6/8 ports through cascade Ethernet port. 5240 provide 2 power connecting methods of power plug and terminal, wide range voltage, as well provide shell grounding protection.

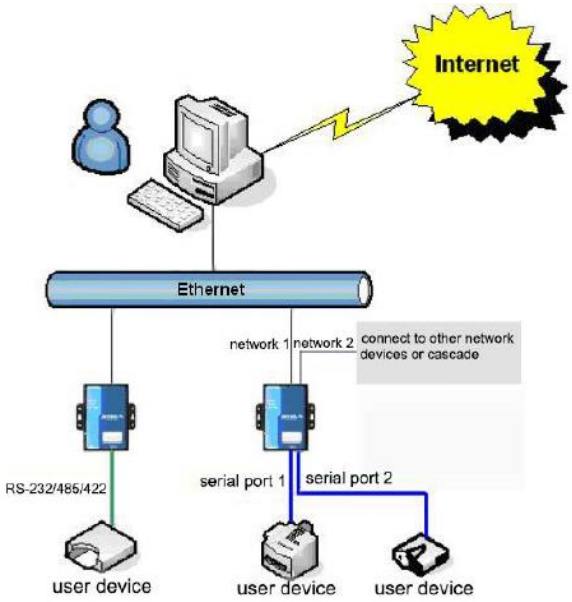
The serial device server can make serial devices conveniently connect to Ethernet and Internet, to achieve network upgrading of serial devices.

ITTLAN5240 is a cost-effective serial device server, RS232 port support full-duplex and uninterrupted communication; RS485 were embedded with 485 lightning protection. Support DHCP/DHS, can easily realize remote device monitoring. Support virtual serial port, the original PC software no need any modification.

Can be applied to:

- Building /Entrance /Door /Security Control System
- Power /Electronic /Intelligent Instrument
- Bank /Medical Automation System
- Stock Exchange System
- Industrial Automation System
- Point-of-sale System (POS)
- Information Household Appliances

Typical application connection is shown as figure 2. First connect the original serial devices with ITTLAN5240, then connect ITTLAN5240 to network via cable. So any data sent by serial devices will be transparently transmitted to PC assigned by ITTLAN5240, and data sent from PC to ITTLAN5240 will be also transparently transmitted to serial devices.



Feature

1. Support full duplex, high speed converting, and no packet lost.
   ITTLAN5240 is 2 ports serial device server of cost-effective, full-duplex, no packet lost. It support simultaneously converting between Ethernet &Serial with large bulk of data with no pause, and also no data is lost.

2. Hight cost performance.
   ITTLAN5240 is designed by concept of intensification, after ensure the stability. It highly takes the cost of networking upgrading in count
3. Support TCP Server, TCP Client, UDP mode, and if communicating with ITTVirCom (our software), it automatically change to Real Com Driver Mode.
4. Support band rate 1200~460800bps, data size 5~9bits, parity of None, Odd, Even, Mark, Space. Support CTS/RTS hardware flow control.
5. Embedded with 485 lightning protection function.
6. Equipped freely with our Windows Virtual Serial & Device Management Tool ITTVirCom. It supports virtual serial and searching device or modifying parameters with ITTVirCom.
7. Provide device management library (Window DLL library). It will help user to develop program with VC, VB, Delphi, C++ Builder. User need only use read() or write() function to communicate with ITTLAN5240.
8. The innovative disconnecting detecting method. Whether it running in TCP Server mode or TCP Client mode, once network is disconnected by some reason, the disconnecting detecting method will detected it and reestablished the connecting.

9. Suitable for Modbus RTU networking upgrading. It is compatible with the software of SCADA/HMI software of Beijing Sunway ForceControl Technology Co. Ltd; Also it support directly transform Modbus TCP to Modbus RTU.
10. With build-in Web server, its parameters can be modified by web browser.
11. Support DHCP, easy for IP management and solve IP confliction.
12. Support DNS. It fulfills the need of access data server through domain name.
13. Support up to 100 TCP connections communicate with network modules at the same time.
14. Flexible serial data framing setting. It fulfills all kinds of serial data frame requirement.
15. UDP mode support dynamic destination address mode. It helps for multi-user mange one serial server.
16. Real Com Driver mode support using the 9-th bit to facilitate communication with milt-device. (the 9-th bit being 0 means data frame and 1 means address frame).
17. Support searching serial servers and modifying parameters through Internet remotely
18. Support parameter modifying protection, preventing modifying by accident. Support running with default parameters.
19. Build-in 2 KV electrical plus protection in RJ45.
20. High protection of electromagnetic interference, with its high electromagnetic interference protection SECC external shell.
21.

Technical parameter:

Figure	➢ Interface: 485: Terminal; 232: RJ45(can equip RJ45 to DB9 cable); 422: Ethernet RJ45 ➢ Power Supply: 5.5mm, Inside positive outside negative, standard outlet; Terminal ➢ Size: L x W x H = 9.4cm x 6.5cm x 2.5cm
Communicate Interface	➢ Ethernet: 2 10M/100M interface (can connect anyone), 2KV electrical plus protection ➢ Serial RS232/485/422×2： RXD， TXD， GND， CTS， RTS
Serial Parameters	➢ Baud rate: 1200~460800bp s Parity: None, Odd, Even, Mark, Space ➢ Data size: 5~9 Flow control: RTS/CTS， NONE
Software	➢ protocol: ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS ➢ Setting method: WEB browser, device management library ➢ Net communication method: Socket, Virtual serial, device management library
Work Mode	➢ TCP server, TCP client, UDP, Real Com Driver
Power:	➢ 9 ~ 24V DC， 2~4W
Environment	➢ Running temperature: - 40~85℃ ➢ Storage temp: - 45~165℃ ➢ Humidity: 5~95%RH
Certifications:	➢ CE, FCC, RoHS

Hardware description

The front view of ITTLAN5103 serial device server is shown as FIG 4: ITTLAN5103 use black anti-radiation SECC board, it's equipped with two "ears" for easy installation.

Size: L×W×H=9.4cm×6.5cm×2.5cm

Panel Lights:

1. ACT: ACT light indicates that data is normally transmitted between Ethernet and rs232/485/422, and when there is no data communication, the ACT light is not on.
2. LINK: the LINK light indicates that the network connection is normal and the TCP connection has been established or in UDP mode.
3. POWER: indicates that the serial device server has already been charged.
4. NET: indicates that the network line of Ethernet1 port has been connected.

The front panel of serial device server is shown in figure above:

(1) Power outlet can adopt standard plug 5.5mm (inner core is positive electrode), voltage 9 ~ 24VDC.

(2) 485 terminal can connect users' 485 positive and negative line.

The back panel of the serial service server is shown in fig. below:

From left to right: the RJ1 is Ethernet1 port, the RJ2 is Ethernet2 port, RJ3 is serial port 1, RJ4 is serial port 2.

(3) RS232 adopts RJ45, and ITTLAN provides a customized RJ45 to DB9 switch, which can be converted to DB9 male head (needle) interface. The line sequence of RJ45 Ethernet port shows as table 1:

Item	Name	Instruction
2	**RXD**	receiving pin of the serial device server
3	**TXD**	sending pin of the serial device server
5	**GND**	Ground wire
1	**RTS**	After the flow control in using, the serial device server will accept the data of the serial device when the pin is 0.
4	**CTS**	After the flow control in using, the serial device server will send the data to the serial device when the pin is 0.

Table 1

The RJ45 to DB9 line, line sequence of DB9 male head shows as table 2:

Ethernet Port No.	Serial Port No.
1	6、8
2	2
3	3
4	4、7
5	5

Table 2

(4) RS422 adopts RJ45 (note that the 422 interface is not opened by default, and the 422 function is required before purchase). The RJ45 line sequence is shown in table 3:

Item	422 line of ITTLAN5240	The corresponding connection line to user RS422
6	T/R+	R+
7	T/R-	R-
8	R+	T+
4	R-	T-

Table 3

Others:

(1) Power supply:

the standard power adaptor (5.5mm inner core positive) or power supply terminal can be used. The power supply voltage can be 9 ~ 24V. The current requirement is greater than 500mA.

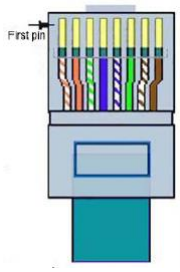
(2) Ethernet port:

A. Ethernet1-normal Ethernet port: the user connects ITTLAN5240 to the switch, hub, or directly to the computer network card via the Ethernet port.

B. Ethernet2-cascade Ethernet port: in the same LAN with Ethernet1, for ITTLAN5240 cascades or connecting with other network devices, referring to the instructions in the document cascade section.

C. POE power supply: through the pin 5 (GND) and pin 8 (VCC) of normal RJ45 Ethernet port supply power to ITTLAN5240, and the power supply voltage is anywhere from 9 ~ 24V.

RJ45 network line sequence reference figure below. By default, POE power is disabled and if need the function please contact ITTLAN.



Cascade Approach

1. ITTLAN5240 support cascade, it's convenient for ITTLAN5240 extending to 4/6/8 ports converter. More convenient for project reconstruction and upgrade.
2. When cascading, use factory equipped cascade network cable (actually shorter parallel network cable) to connect cascade Ethernet port (Uplink RJ45) of up level ITTLAN5240 with common Ethernet port (RJ45) of next level ITTLAN5240. The cascade steps can be done continuously, up to 8.
3. In default every level of ITTLAN5240 needs power supply. If need power supply through cascade cable, should be customized.

485 Character

ITTLAN5240 meet the RS485 standard, each ITTLAN5240 can be with 32 terminal 485 devices. The maximum communication distance is 1200 meter, the resistance of 485 terminal is 120 ohms, usually must use terminal resistance when wiring over 300m. Pay attention to the wiring, 485+ and 485- must be a twisted-pair, in order to reduce signal interference.

Usage

8.1 Hardware connection

Generally speaking, serial port server only needs to connect power supply, serial port and network cable.

The power supply can be configured with a 12V power adapter or a 2-wire power supply on site, which can be directly connected to the positive and negative terminals of the power supply.

RS485 device 485 is connected to TA, 485 is connected to TB. The network port is connected to the ordinary network cable, which can be directly connected to the computer or connected to the network through the switch.

8.2 Software Installation

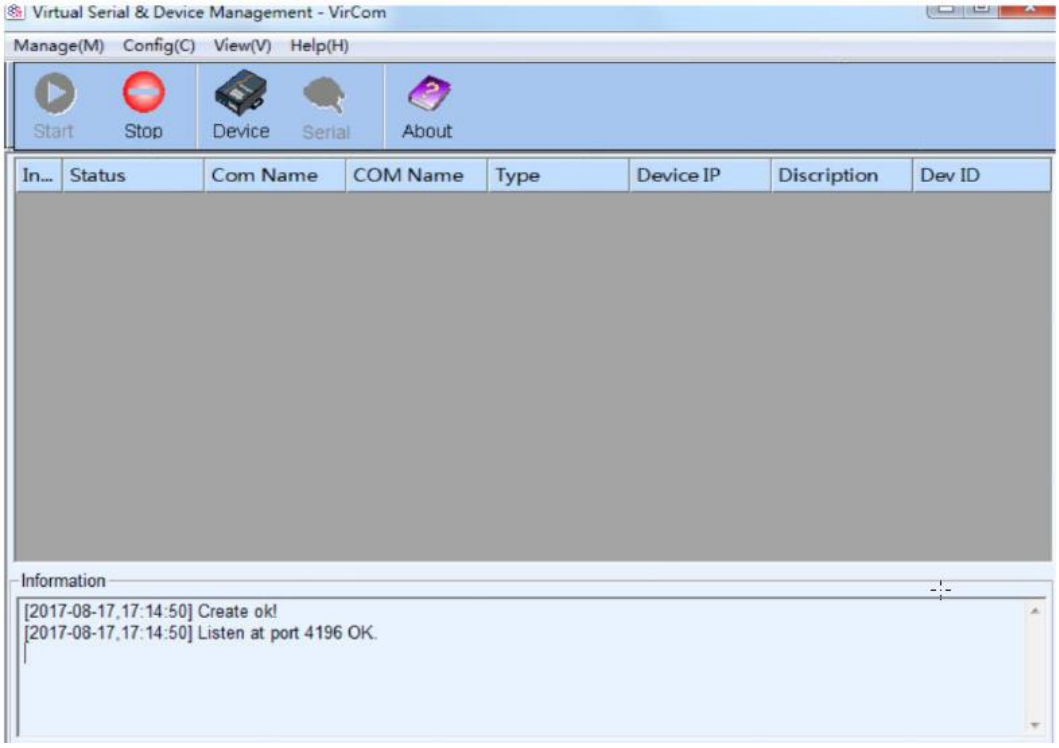
ITTVirCom can be used to configure the device IP and other parameters, also can create virtual serial port. If no need the virtual serial port function, you can download the free-installation version.

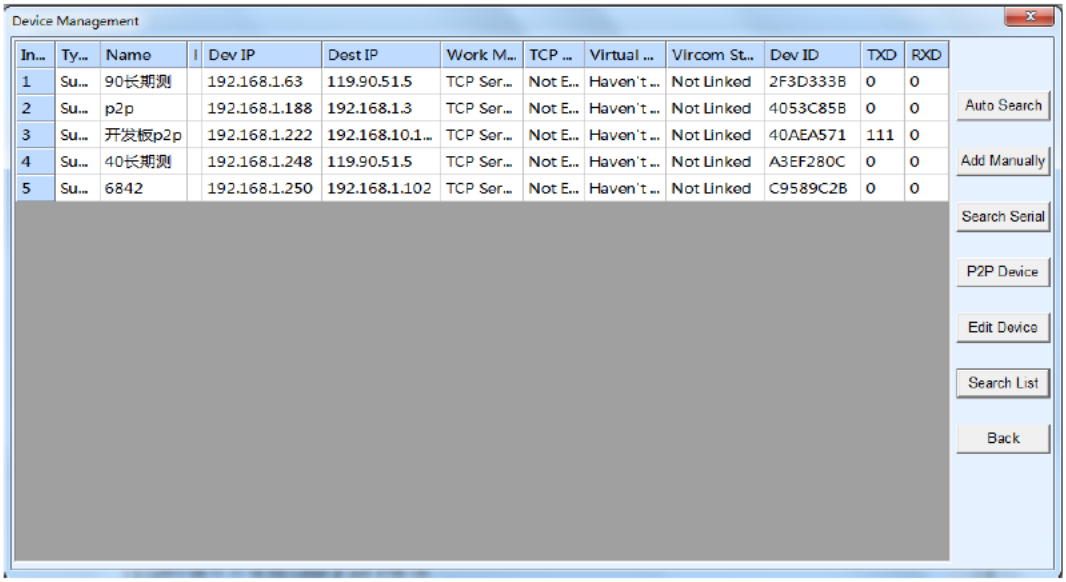
Software	Description
ITTVircom Device Management Tool (free-installation version)	This version no have the virtual serial port function
ITTVircom Device Management Tool (installation version)	This version has the ITVircom_x64.msi and ITVircom_x86.msi. The 64 bit operating system install x64, 32 bit system install x86 version.

Just follow the default prompt when installing. Upon completion of the installation, ITTVirCom will be started each time the computer is started, which is used to boot up to create a virtual serial port.

8.3 Parameter Configuration

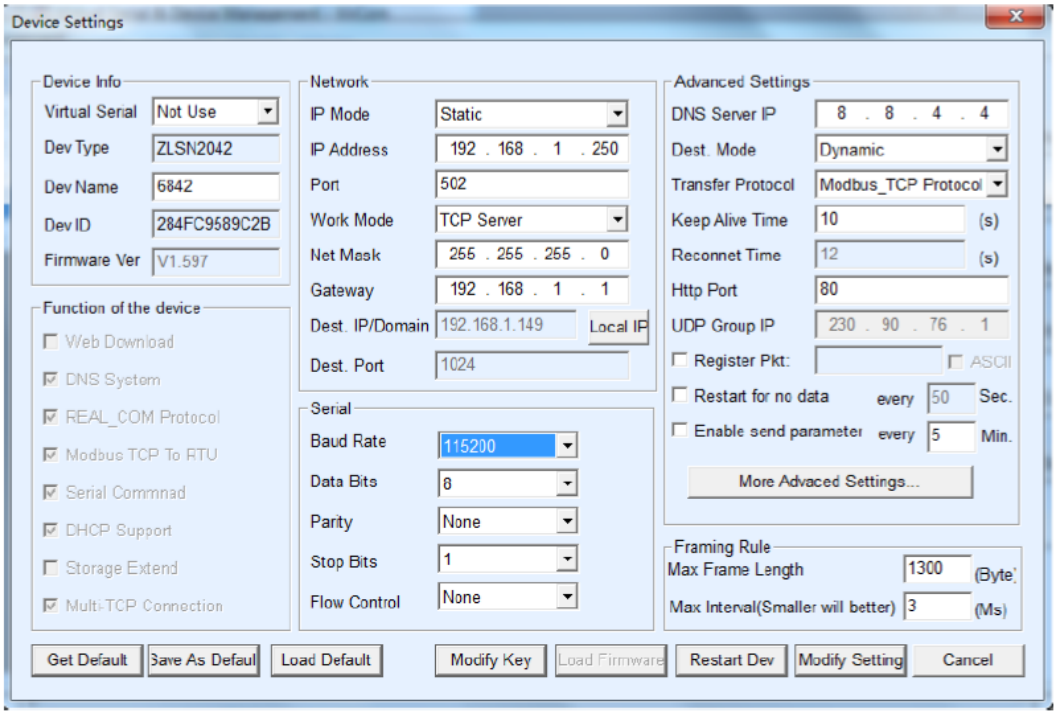
After installing ITTVirCom, the hardware also connecting, run ITTVirCom software as figure 7, and click "Device Manage" as figure 6. Use ITTVirCom can search and configure the device parameter in different segment, which is very convenient as long as the device and computer of running ITTVirCom are under the same switch.





From the device list, you can see all of the current online devices, and you can search for devices that are not in one network segment. There is no need to use the "Add Manually" function.

Click the “Edit Device” to configure the device parameters.



In this interface, the user can set the parameters of the device, then click "Modify Setting", and the parameters are set to the flash of the device, with power-off no lost. The device will restart automatically.

The main configuration parameters are: baud rate, data bit, and check bit in serial port setting; IP address, subnet mask, gateway in network setting; Sometimes you need to configure the work mode of the serial port server according to the computer software.

The meaning of the parameters is described as follows:

Parameter Name	Value Range	Instruction
Virtual Serial	Non-in use, created virtual serial port	You can bind the current device to a created virtual serial port.
Dev Type		Show only the model of the core module

Dev Name	Any	You can give the device a readable name, with a maximum of 9 bytes, and support the Chinese name.
Dev ID		The factory's sole ID, cannot be modified.
Firmware Version		The firmware version of core module
Supporting Function		Please refer to the Table 4 “support functions”
IP Mode	Static, DHCP	The user can choose Static or DHCP (Dynamic acquisition of IP)
IP Address		The IP Address of networking products
Port	0～65535	The monitoring port of Networking products when in the TCP Server or UDP mode. As a client, it is best to specify that the port is port 0, which is good for increasing the connection speed, and the system will randomly assign a local port when using the 0 port. At this time the difference from specifying the non-zero port are: (1) local port is 0, module sets up a new TCP connection with PC when restarting, old TCP connection may not be closed, so that the old TCP connection of the host has been unable to close, specify the non-zero port does not have the problem. Generally host wants to close the old connection when the module is restarted. (2) the local port is 0, the time of TCP rebuilding connection is faster.
Work Mode	TCP Server(TCP Server Mode),TCP Client(TCP Client Mode),UDP Mode, UDP Multicast	When set to TCP Server, the network Server needs to actively connect the networking products; When set to TCP Client, the networking product initiates the connection to the network server specified by the destination IP.

Net Mask	Eg: 255.255.255.0	Must be same as net mask of local LAN.
Gateway	Eg: 192.168.1.1	Must be the same as the local LAN gateway. If it is not crossing outer network (such as the cable connecting computer), it is best to set the gateway as the IP address of the connected computer.
Dest. IP/Domain		In the TCP Client or UDP mode, the data will be sent to the destination IP or the computer of domain name instruction.
Dest. Port		In the TCP Client or UDP mode, the data is sent to the destination port of the destination IP.
Baud Rate	1200, 2400, 4800, 7200, 9600, 14400, 19200,	Serial baud rate
	28800, 38400, 57600, 76800, 115200, 230400, 460800	
Data Bits	5, 6, 7, 8, 9	
Parity	None, Even, Odd, Mark, Space	
Stop Bits	1, 2	
Flow Control	None (no flow control), CTS/RTS, DTR/DCR, XON/XOFF	RS232 port valid
DNS Server IP		When the destination computer is described by a domain name, DNS server is required to resolve the domain name, which specifies the IP of this DNS server. When the IP mode is DHCP, the parameter is not specified and will be automatically acquired.

Dest. Mode	Static, Dynamic	UDP working mode: if the destination computer is described by a domain name, it's best to choose the static mode; If there are multiple computers in the LAN communicating with networking products through UDP, it is best to choose dynamic mode. TCP server mode: this parameter must be dynamic. TCP client mode: when IP mode is dynamic, the destination IP is reconnected after the device is restarted, so that the correct IP address can be obtained again. Otherwise, it will do direct connection without automatically restarting the device.
Transfer Protocol	NONE, Modbus TCP<->RTU, Real_COM	NONE indicates that the data forwarding from the serial port to the network is transparent; Modbus TCP<->RTU will convert Modbus TCP protocol directly into RTU protocol to facilitate coordination with Modbus TCP protocol; RealCOM is designed to be compatible with the old version of REAL_COM.
Keep Active Time	0～255	(1) Choose 1~255, if the device is in the TCP client working mode, the TCP heartbeat will be sent automatically for every "keep alive time". This can guarantee the TCP availability of links. When set to 0, there will be no TCP heartbeat. (2) Set to 0~254, when transformation protocol choose REAL_COM protocol, the device will send a length of 0

		to 1 content data for every " keep alive time " to implement the heartbeat mechanism of Realcom. When set to 255, there will be no Realcom heartbeat. (3) Set to 0~254, if the device is working on the TCP client, the device will send the parameters to the destination computer every " keep alive time ". When set to 255, no have the parameter sending function. This mechanism is not normally used, users are not required to pay attention.
Reconnect Time	0～255	Once the networking products in a TCP client mode disconnect with the server (as long as in the non-connection status), it will initiates a TCP connection to the Server every while, can be 0~254 seconds, if set 255, never for reconnection. Note first TCP connection would immediately (such as hardware on electricity, through ittvircom software restart equipment, no data), only after the first connection failure will try again after waiting for the "break time", so "break time" will not affect the network and server connection setup time under normal circumstances.

Http Port	1～65535	
UDP Group IP		UDP multicast
Max Frame Length	1～1400	One of the rules of serial. The connected product serial port sends the received data to the network as a frame after receiving the length data.
Max Interval (Smaller will better)	0～255	One of the rules of serial. When there is a pause in the data received by the connected product, and the pause time is greater than that time, the received data is sent to the network as a frame.

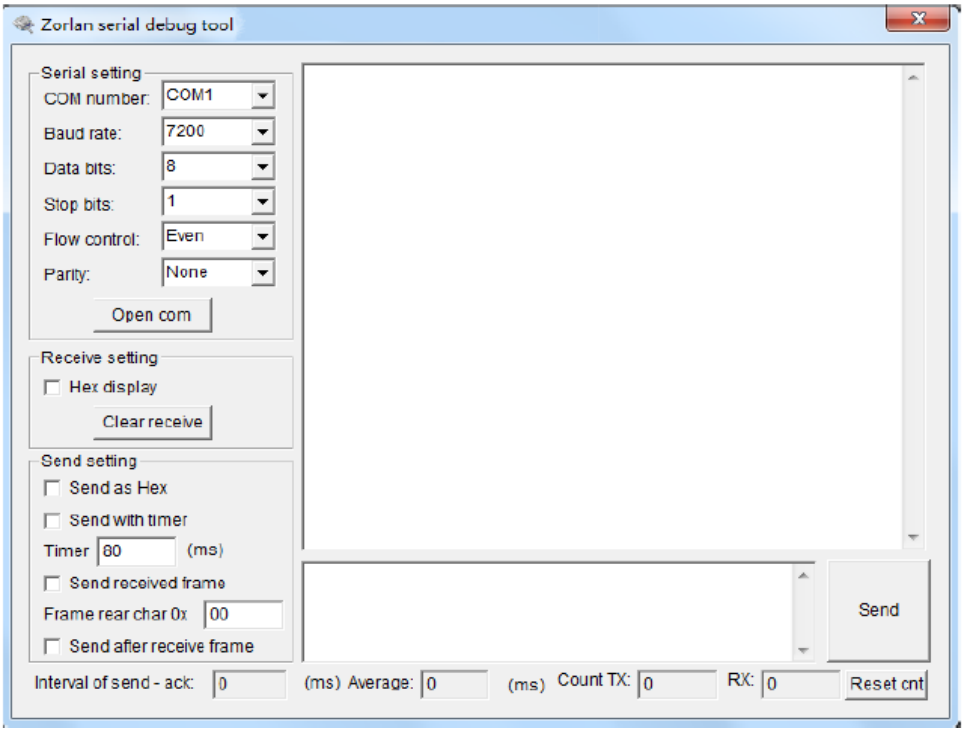
Name	Instruction
Webpage download	Support to control the serial output command through the web page, only the product with the trailing W has this function.
Domain name system	The destination IP can be the domain name (such as the beginning of the WWW server address).
REAL_COM protocol	It is a non-transparent transmission serial device server protocol,
	which is suitable for the binding of virtual serial port through Internet. Because the protocol contains the device MAC address, it is helpful for the upper computer to identify the device. In general, it can be without use.
Modbus TCP to RTU	Only the models that the third bit is 4 support this function. Modbus TCP to RTU can be implemented. It also supports multi-host capabilities.
Serial port modify parameters	The serial port modification parameter supports the serial port AT instruction to configure and read the device parameters.
Automatic acquisition of IP	Support for DHCP client protocol
Storage extension EX function	Extended later
Multiple TCP connections	Support more than one TCP connections as a TCP server.
IO port control	Model No. 3 of 4 supports arbitrary custom instructions to control 8 IO outputs.
UDP multicast	UDP multicast
Multi-destination IP	As a TCP client supports simultaneous connection of 7 destination IP.
Proxy server	Support the proxy server functionality (a specific model is required).
SNMP function	Support SNMP to Modbus RTU protocol. Only those with -snmp tails support this feature.
P2 function	Support the ability to access devices in any network through P2P across technology. This feature is supported in model N with trailing patches.

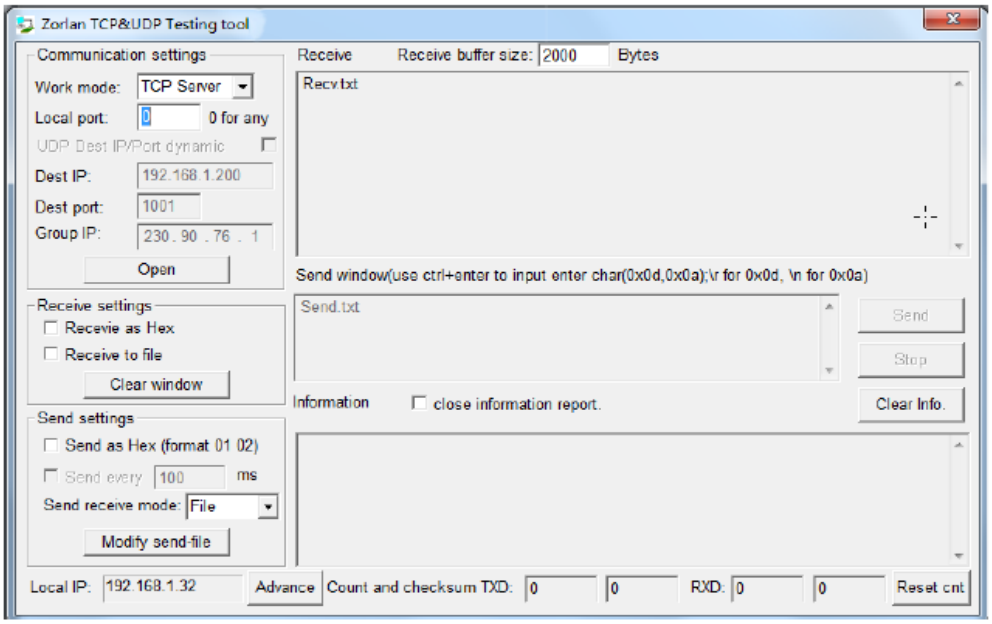
8.4 TCP communication test

After the device parameters are configured, TCP connection communication can be tested with serial port tools and TCP debugging tools.

Now suppose the PC COM port (USB to RS232 line) connect with serial port of serial device server, then open the serial debugging assistant ITTComDebug, and open the corresponding COM as figure 9; Open TCP&UDP debugging assistant SocketTest, and as TCP client, fill in the destination IP for the serial port server IP (currently 192.168.1.200), destination port is 4196, and then click "open" button in figure 10. Enter "socket send" in SocketTest and click send, then the data is transferred to RS232 interface through the network port of the serial port server, and then sent to ITTComDebug, which is then displayed in ITTComDebug. Conversely, type "Comdebug send" in ITTComDebug, and clicking send can also be sent to socket test and displayed.

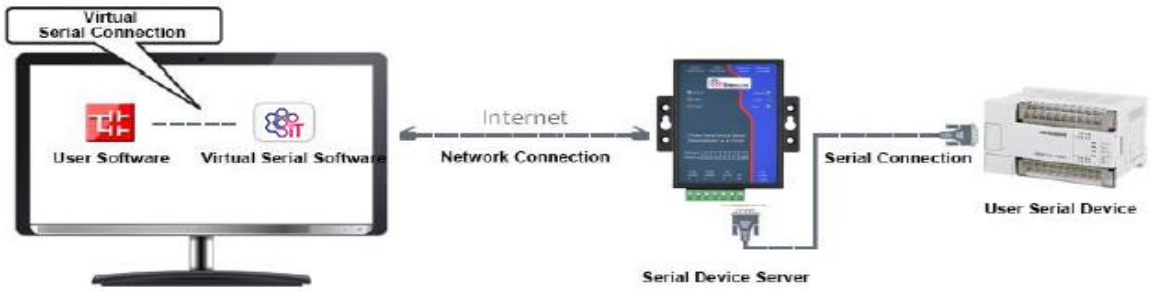
The demonstration demonstrates the data transparent forwarding function from serial port to network port, network port to serial port.



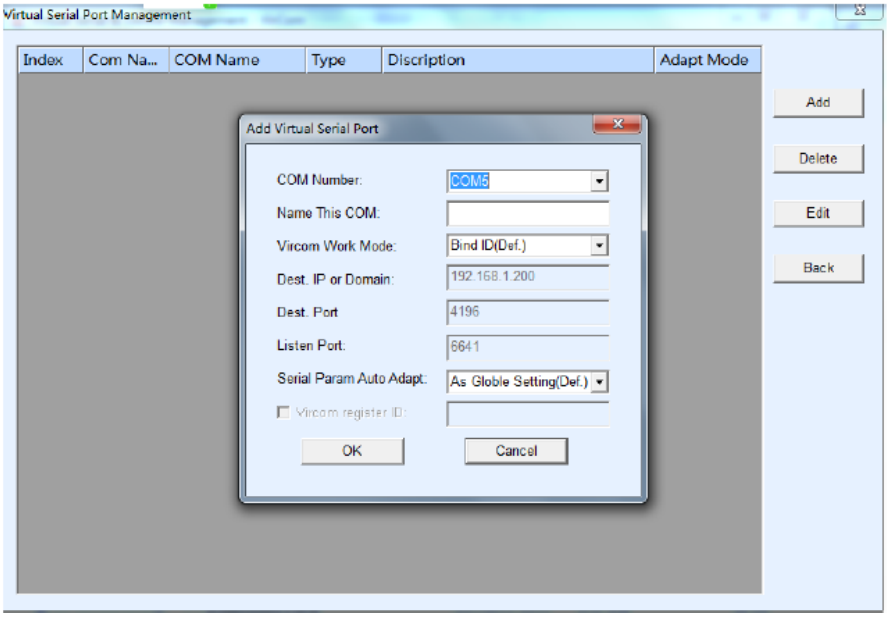


8.5 Virtual serial port test

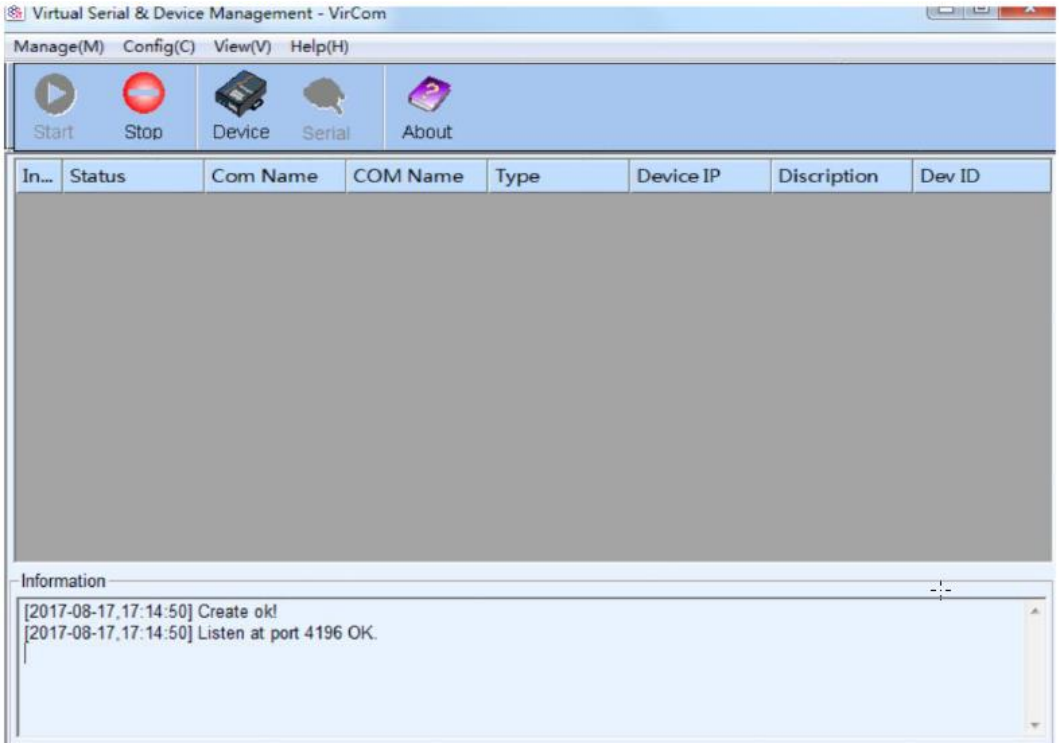
SocketTest in figure 8 communicates directly with the serial port server through TCP. In order to enable users to communicate with the serial port server with the developed serial port software, it is necessary to add a virtual serial port between the user program and the serial port server. As shown in figure 11, ITTVirCom and the user program run on a computer, and ITTVirCom virtual a COM port, so that the COM port corresponds to the serial port server. When the user program opens the COM to communicate through ITTVirCom □ serial device server □ to users. Here's how to do this:



Click on the "Serial Manage" of the main interface of ITTVirCom, then click "Add", and select COM5, where COM5 is the COM port that didn’t exist on the computer.

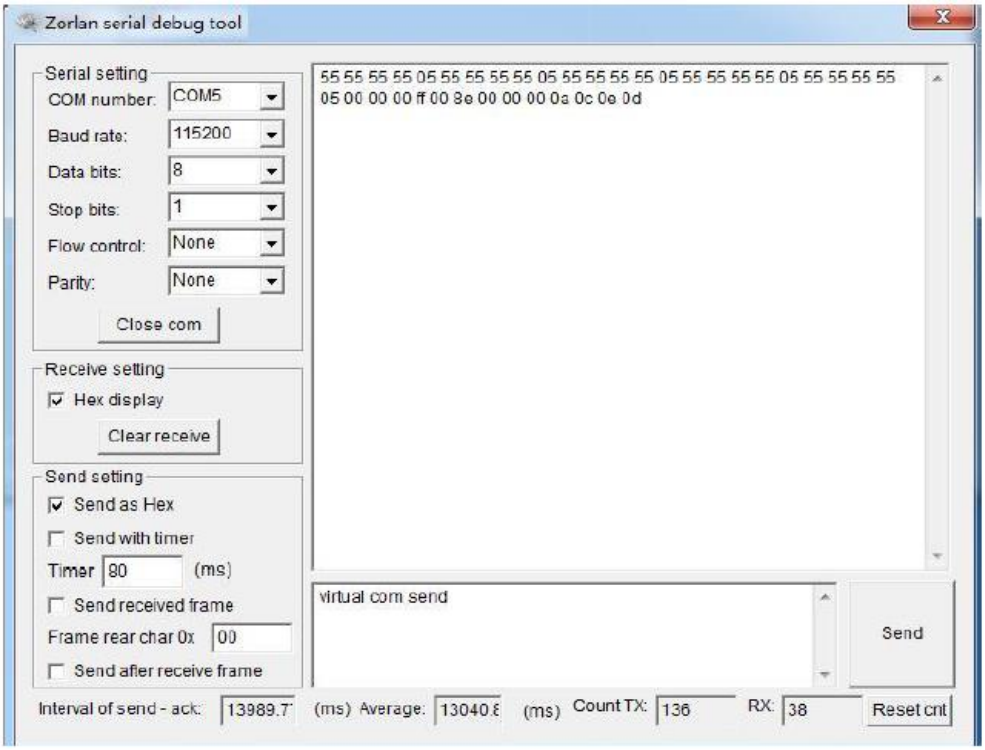


Then enter the “device manage”, and double-click the device that you need to bind to the COM5. As shown in FIG. 9, select COM5 from the "virtual serial port" list in the upper left corner. Then click "modify Settings". And return to the main interface of ITTVirCom. You can see that the COM5 has been connected to a device with IP 192.168.1.200. You can use COM5 instead of SocketTest to communicate.



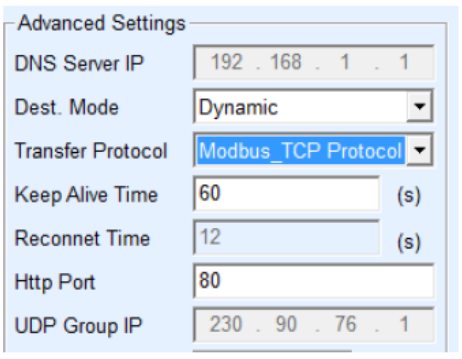
Now close the before SocketTest and open a new ITTComDebug as the user's serial port program, now open COM5. At this point, COM5 (virtual serial port) and COM4 (hardware serial port) can send-receive data through networking products. If the serial port of the connected

product is not connected to the COM port of PC, but a serial port device, then the COM5 can be opened to communicate with the device. And it's just use the network way now.

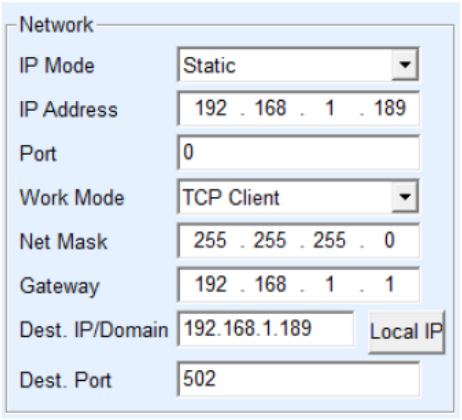


8.6 Modbus TCP test

By default, serial and network data is transmitted transparently. If you need to implement the Modbus TCP to RTU, requires selecting “transfer protocol” for "Modbus TCP -> RTU" in the Settings dialog, as shown in figure 20. At this point, the device port is automatically changed to 502. At this point, the user's Modbus TCP tool is connected to port 502 of IP of serial port server, and the Modbus TCP instruction sent will be converted into RTU instruction output from the serial port. For example, the serial port server network port receives the Modbus TCP instruction of 00 00 00 00 00 06 01 03 00 00 0a, then the serial port outputs the instruction of 01 03 00 00 00 0a c5 cd. Note: the serial port may send several 01 03 00 00 00 0a c5 cd instructions, because the default Modbus adopts the storage mode, which will automatically poll to query. It'll show you how to switch to non-storage.



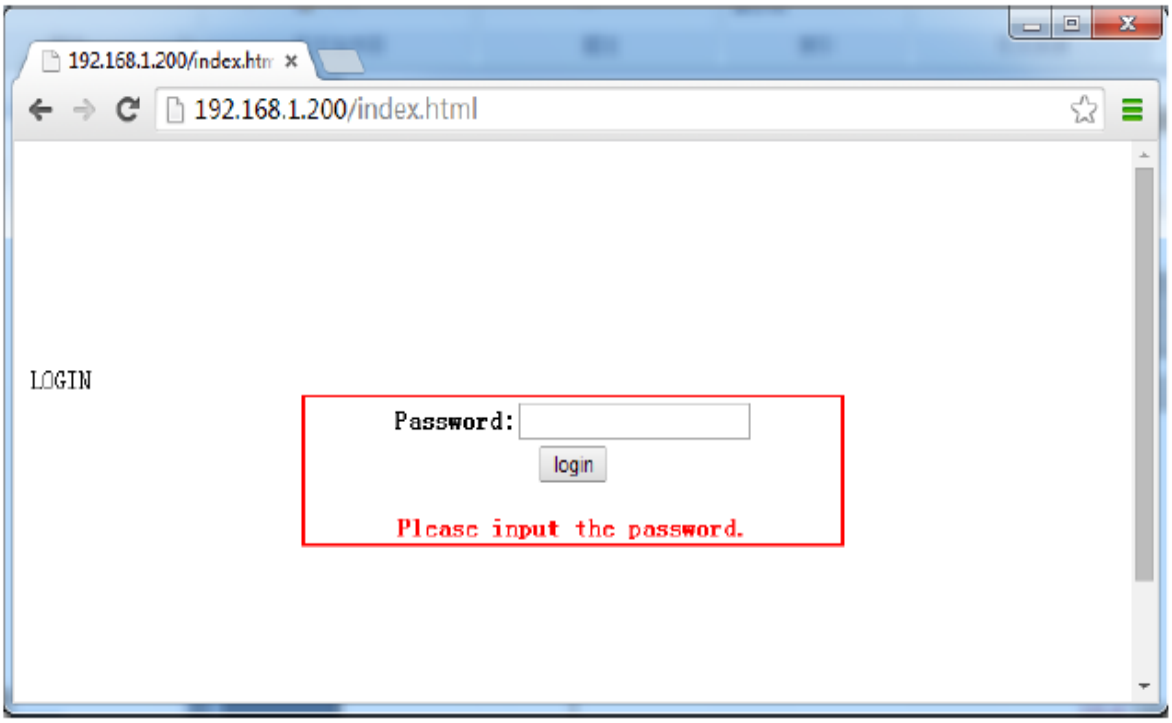
If the user's Modbus TCP software is used as Slave, it is necessary to change the working mode to client based on the transfer protocol selection, and the destination IP change to the computer IP of Modbus TCP software, and the destination port is 502, as shown in figure 18.



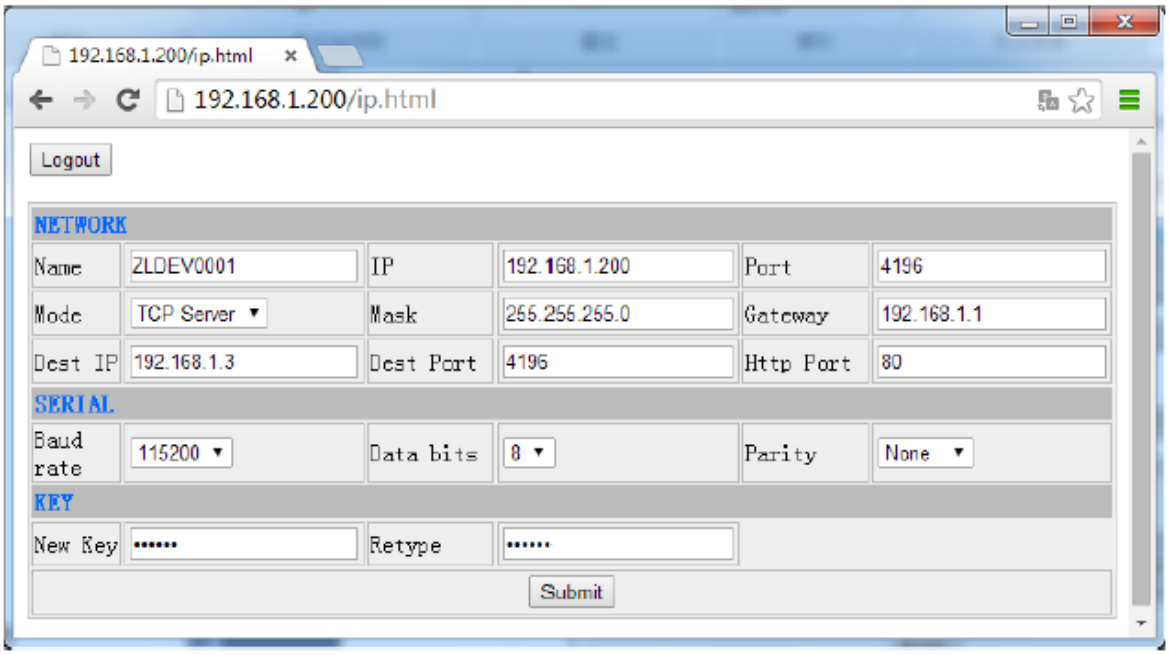
8.7 Web Configuration

ITTVirCom can be used to search and configure device parameters in different network segments. The configuration of the Web mode requires that the computer and the serial port server are in the same IP segment, and the IP address of the serial port server should be known in advance. But the Web configuration can be done on any computer without ITTVirCom.

Enter the IP address of the serial port server in the browser, such as http://192.168.1.200, to open the following page.



Enter Password in Password: default is 123456. Click the login button to login.



The serial port server parameters can be modified in the web page that appears, and the related parameters can refer to the meaning of the parameters in table

After modifying parameters, click the "submit" button.

8. Work Mode and Transfer Protocol

In different applications can choose different work mode, transfer protocol of the serial server, so as to be more stable and reliable use, the details are as following.

The use of serial device server is basically divided into two types: with virtual serial port and non-virtual serial port, as shown in fig.11 TCP communication diagram and fig.16 function of virtual serial port. The user software with virtual serial port needs to be connected is the COM port, that is, both the user software and the user device are serial ports. Non-virtual serial mode user software communicates directly over TCP/IP but user devices are still serial.

In the non-virtual serial port mode, the "Transfer Protocol" is divided into transparent transmission, Modbus TCP to RTU and Realcom protocol. If the user software is a fixed Modbus TCP protocol and the host computer is Modbus RTU, Modbus TCP to RTU mode should be selected. The Realcom protocol is currently only used when a multi-port server is connected to a server as a TCP client and a virtual serial port is used on the server.

Usage is summarized as follows:

No.	Virtual Serial Port Usage	Device Work Mode	Transfer Protocol	Instructions
1	Use	TCP Server	None	Suitable for the user to open the COM port to collect data actively.
2	Use	TCP Client	None	Suitable for the occasion when the device sends data actively. If the TCP server is selected, the problem that the device cannot be reconnected after being disconnected may occur.
3	No Use	TCP Server	Modbus TCP to RTU	The user software is Modbus TCP, and the user device is Modbus RTU. Modbus TCP is the main station.
4	No Use	TCP Client	Modbus TCP to RTU	The user software is Modbus TCP, and the user device is Modbus RTU. And Modbus RTU is the main station.
5	Use	TCP Client	Realcom Protocol	When using a multi-port server as a TCP client and the virtual serial port is used, it is best to use the Realcom protocol.
6	No Use	TCP Client	None	Suitable for a large number of devices, connected to a cloud way. And under normal circumstances the cloud is a server of public network IP on the Internet.
7	No Use	TCP Server	None	Suitable for devices and computers in the same local network, do local monitoring, no need to communicate across the Internet.

8.1 Virtual Serial Mode

If the user software is using COM port for communication, it must use virtual serial port mode. Including some PLC software, configuration software, instrument software, etc.

Then check whether the monitoring computer and equipment are in the local network:

1. If the computer is a public network IP server rented over the Internet, then the device must use TCP client mode to connect the device to the server. At this point, you can select the (2) and (5) in Table 6, and must select (5) if it is a multi-port server.
2. All in the local network (can ping each other), then see is the upper computer active query or device active send data. If the device is sent initiatively it must use the device to be TCP client as (2) in Table 5, or you can choose the (1) way.

8.2 Direct TCP/IP Communication Mode

If Modbus TCP conversion is not required and virtual serial port is not required, the user software may directly communicate with the network port of the serial port server for TCP/IP communication, and the serial port server will transfer the TCP/IP data to the serial port data and send it to the serial port device.

Generally, users of this kind of usage develop the network communication software of upper computer by themselves and integrate the analysis of the serial communication protocol of the equipment. This method is more flexible and efficient than virtual serial port. Correspond to (6) and (7) in table 6.

In the section of "6.4 TCP Communication Test", it mainly introduces how to communicate when the serial port server is a TCP server. Here you will learn about TCP clients, UDP mode, and how multi-TCP connections communicate with computer software. The computer software takes SocketTest (the software that imitates the user's TCP/IP to communicate) as an example.

ITTLAN serial port server is in compliance with the standard TCP/IP protocol, so any network terminal that conforms to the protocol can communicate with the serial port server. ITTLAN technology provides the network debugging tool (SocketDlgTest program) to simulate the network terminal to communicate with the serial port server.

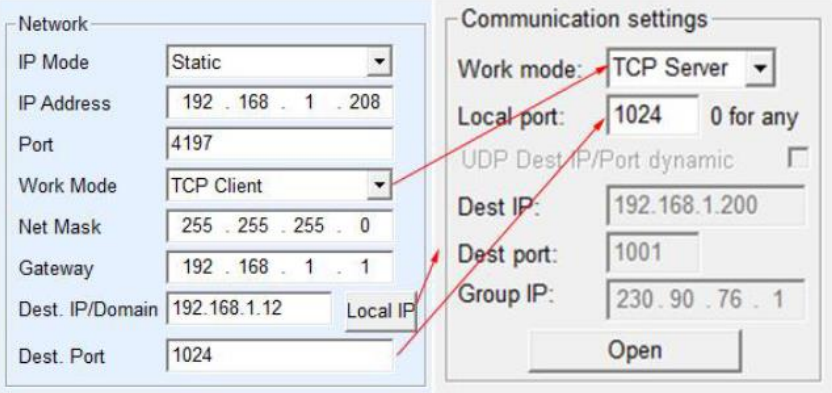
For two network terminals (in this case, the network debugging tool and the serial port server) to be able to communicate, the configuration of the parameters must be matched.

8.1.1 TCP Client Mode

There are two operating modes under TCP mode: TCP Server and TCP Client. No matter which mode is adopted, one side must be the Server side and the other side the Client side. Then the Client side can access the Server side.

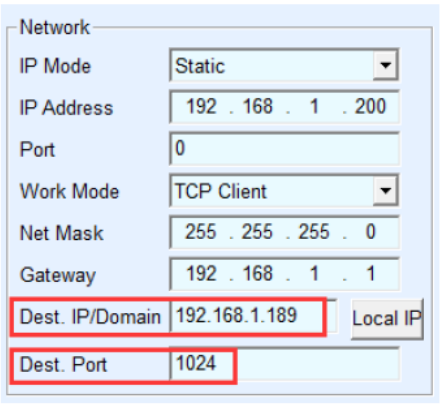
When the serial port server is the Client, there must be three corresponding relationships, as shown in figure 24. 1) working mode correspondence: the working mode of the serial port server is the Server mode corresponding to the Client mode of the network tool; 2) IP address correspondence: the destination IP of the serial port server must be the IP address of the computer on which the network tool is located; (3) port correspondence: the destination port

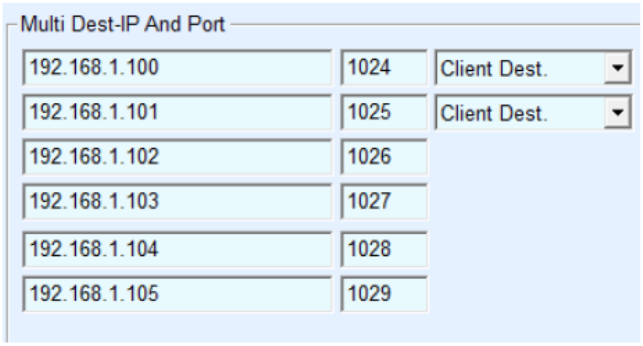
of the serial port server must be the local port of the network tool. In this way, the serial port server can automatically connect to the network tools, send and receive data after the connection is established.



8.1.2 The Client connects to Multiple Servers

When the ITTLAN serial port server is a TCP client, it can connect to 7 destination IP addresses at the same time, and the data sent by the serial port will be sent to 7 destination IP addresses at the same time. If you don't have that many servers, the rest of the destination IP is vacant. Its use method is as follows:





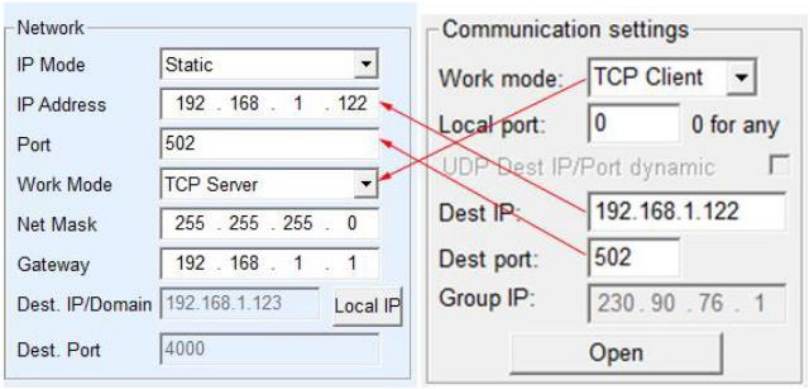
The first IP is set in the device settings interface shown in figure 25, where the first IP can be the domain name. For the remaining 2~7 destination IP, click the "More Advanced Options" button in the device setting interface and open more advanced options for setting.

All 7 destination IP Settings can be automatically connected, if not connected, will wait for "disconnect reconnect" time after repeated reconnect.

Note that if the wifi parameter is set, the number of multi-destination IP will be reduced (because the wifi parameter is temporarily used in the parameter configuration table). If you still need to maintain multiple destination IP and need to configure the wifi parameter, please use the format of 18.2wifi.txt file to configure the wifi parameter, and leave the parameter table empty.

8.1.3 TCP Server Mode

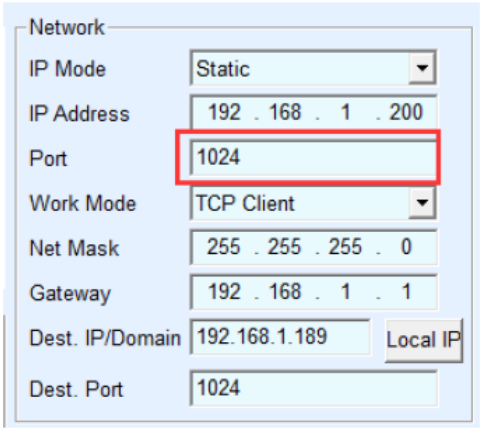
When the serial port server acts as a Server, there are also three corresponding relationships, as shown in figure below. There is no explanation here. After this setting, click the open button of the network tool to establish a TCP connection with the serial port server, send and receive data after the connection is established.



When the serial server is as Server, it can accept 30 TCP connections simultaneously. The data received by the serial port is forwarded to all established TCP connections. If the data need to be sent only to TCP that has recently received network packets, multi-host function needs to be enabled. Please refer to “9.4 Multi-host Function”.

8.1.4 Be both Client and Server

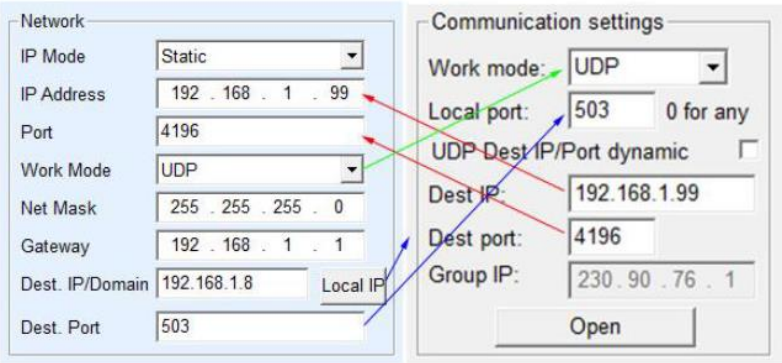
The ITTLAN serial server supports accepting TCP connections when the device is on a TCP client, which means that it also has TCP server functions.



By default, when configured with ITTVirCom, if the working mode is changed to "TCP Client" mode, the port (that is, the local port) automatically becomes 0 (0 means a random free port is selected). In order to support the TCP Server mode, the computer software must know the local port of the device, so a value needs to be specified. As shown in figure 23, the computer software can now connect to port 1024 of 192.168.1.200 for communication, and the device will also connect to port 1024 of 192.168.1.189 as the Client. It should be noted that since the local port 1024 is occupied by the Server, when the local port is used as the Client, it is "port +1", that is, the software on 192.168.1.189 sees that the incoming port of the device is 1024+1=1025.

8.1.5 UDP Mode

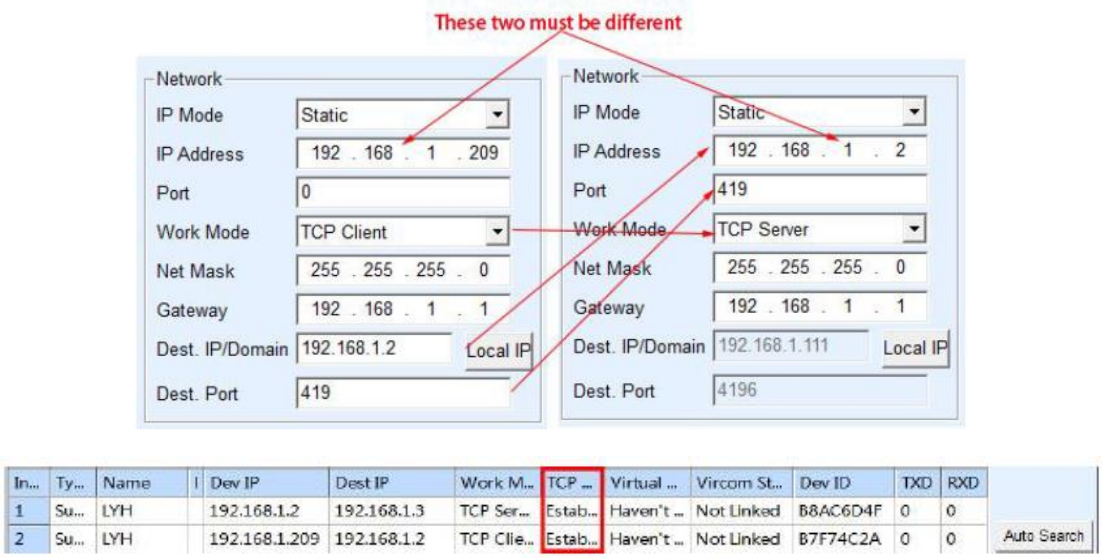
In UDP mode, the parameter configuration is shown in figure 24. The left side is the configuration of the serial port server in ITTVirCom, and the right side is the setting of the network debugging tool SocketDlgTest. First, both must be UDP working modes. In addition, as indicated by the red arrow, the destination IP and destination port of the network tool must point to the local IP and port of the serial server. As indicated by the blue arrow, the destination IP of the serial port server must be the IP address of the computer where the network tool is located, while the destination port of the serial port server must be the local port of the network debugging tool. Only when these network parameters are configured the two-way UDP data communication can be guaranteed.



8.3 Device Pairs-connect Mode

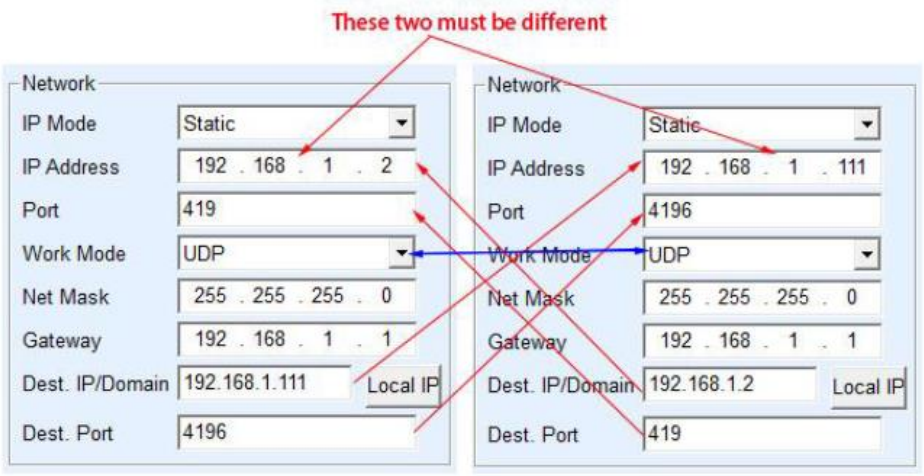
If the upper machine is not a Socket program (SocketDlgTest) or a virtual serial port, but two devices are connected through a network port, the configuration method is similar. First, the user needs to connect two devices and computers to the same LAN. ITTVirCom runs on this computer and is only connected to the computer for configuration purposes, after which the computer does not need to be connected.

Click ITTVirCom's device management to find the two devices, as shown in figure 26. Then click "Device Edit" to configure the device. Device pair-connection can be divided into TCP pair and UDP pair. In the case of TCP pairing, the parameters of the two devices are shown in figure 25. The parameters shown by the arrows must correspond as if they were connected to a PC. After a successful TCP connection, you can view the connection status by going back to the device management dialog box, as shown in figure 26. If both devices are "connected", the TCP link for both devices has been established.



In the case of UDP pair-connection, the configuration parameters are shown in figure 27, and the parameters corresponding to the arrows must be one-to-one. As long as the parameters of

UDP pair are configured correctly and the connection status is not checked, the data sent will be automatically sent to the specified device.



Finally, we need to remind that if the device is connected to each other, in addition to the above settings of the Ethernet port parameters, the serial port parameters must also be correctly set. Mainly is the serial port server baud rate and so on needs to be consistent with user's equipment baud rate and so on. With this setup, the user device can send data to each other through the serial ports of the two serial servers.

9. Device Debugging

9.1 Network Physical Connection

Confirm the wifi connection is normal through wifi_work and wifi_link indicator lights.

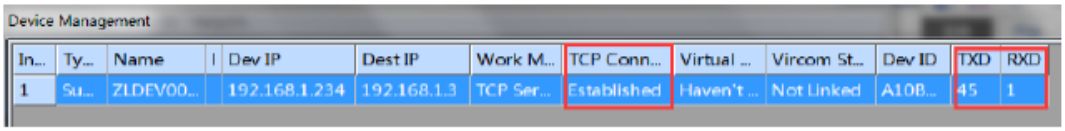
9.2 Network TCP Connection

When the device is in dynamic accessing IP mode, cannot directly connect to the computer network interface. Because no have DHCP server to be able to use (general DHCP server is the router in LAN). So please specify IP when connecting directly. The computer also needs to specify a fixed IP address.

When configured with static IP, both the device and the computer need to be on the same network segment (unless communicating across gateways), either directly or through a switch, as shown in figure 33.

Since ITTVirCom supports cross-segment search and configuration, the IP address that can be searched but cannot be communicated is usually not well configured. In this case, ITTVirCom can be used to configure devices in the same network segment.

After the configuration, the Link light turns blue when the TCP connection is established by using the steps of “6.4 TCP Communication Test” or “6.5 Virtual Serial Port Test”. The Link light blue can also be seen through ITTVirCom, as in the device management list, if the TCP connection is listed as "established" then the Link light is blue, as shown in figure 34, which facilitates remote diagnosis.



9.3 Data Sending and Receiving

When the Link light turns blue, data can be sent and received between the software and the serial port server. If the software sends a message, the Active light turns green for at least a second. The data is also output from the serial port server, but whether the output data is correct depends on whether the correct serial port parameters (baud rate, data bit, stop bit, check bit) are configured.

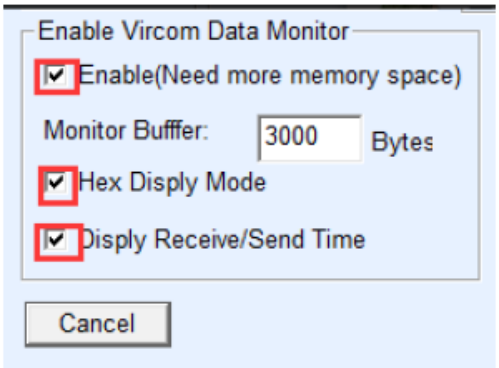
The serial device will reply to the right instruction. If there is a reply (the serial port sends data to the network port), “Active” will turn blue. Otherwise, please check the serial port parameters or the connection of the serial port line.

In order to facilitate remote debugging, ITTVirCom also supports remote viewing of data sent and received, as shown in figure 29, where TXD is the amount of data sent by the serial port server. When refreshing the list of devices, a change in this value indicates that data has been sent, and the Active light is also green. If you see a change in the RXD value it indicates that the serial device has returned data. Active is blue.

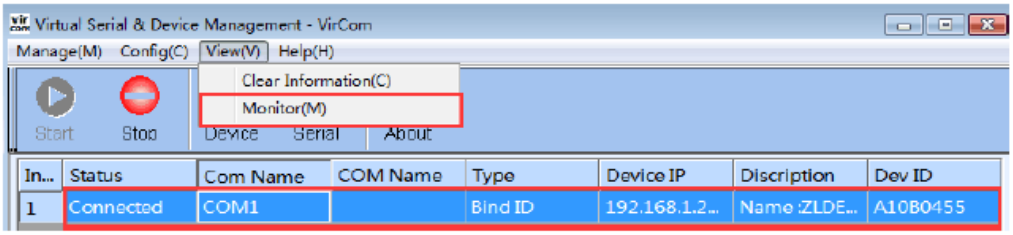
9.4 ITTVirCom Remotely Monitor Data

In the case of virtual serial port, ITTVirCom supports real-time capture of data sent and received from the virtual serial port. It is convenient for users to debug the system. The use method is as follows:

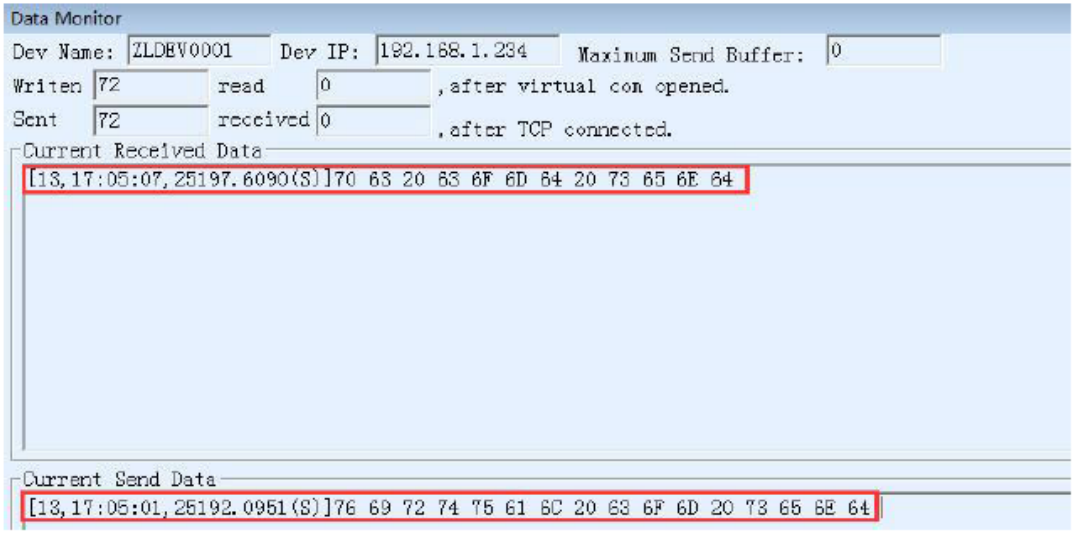
It is assumed that the virtual serial port communication has been established according to the "6.5 Virtual Serial Port Test" method. Now you need to monitor the data passing through the virtual serial port. Open the ITTVirCom menu/configuration/software configuration/open the vircom configuration dialog box.



Select the three options of enable monitoring, hexadecimal monitoring mode, and display the time of send-receive data, as shown in figure 31. Then click ok. Assuming that data has been sent and received before, now select a virtual serial port to be monitored in the main interface, and then select the menu/view/monitor, as shown in figure 36.



From the open dialog box, you can see the instructions issued by the upper computer and the instructions returned by the device, as shown in figure 32. This function can be convenient for field communication debugging.



10. Modbus Advanced Features

The serial port server with Modbus gateway function does not have station address and register. It is a communication bridge. It will generate Modbus RTU designation according to Salve ID, function code, register No. and register number in Modbus TCP instructions sent to Modbus gateway by user software and output them from the serial port. Think of it as a protocol "translator."

10.1 Enable Modbus Gateway

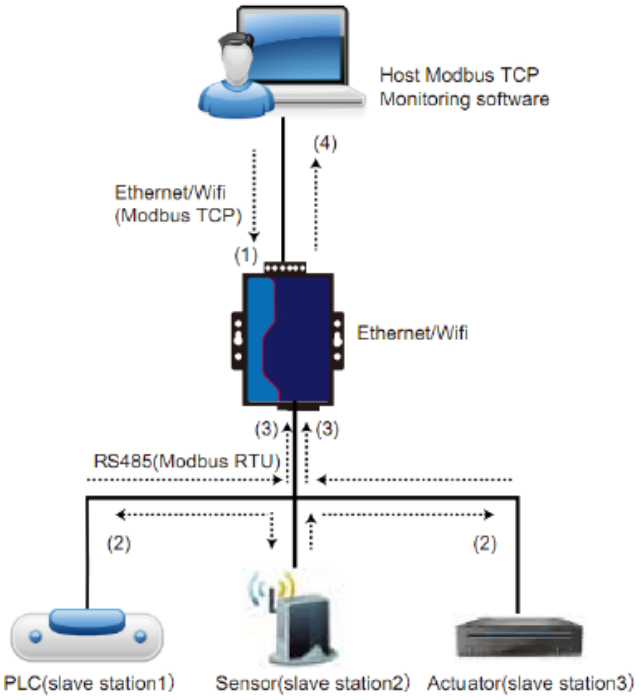
First of all, the serial port server should support Modbus gateway, that is, the function of "Modbus TCP to RTU" supported by devices in table 5 in the device Settings dialog box should be checked.

By default, a serial port server is in ordinary transparent transmission mode, if you need to Modbus gateway pattern, please choose the option "Modbus TCP??RTU" in the "transfer protocol". The device then automatically changes the "port" parameter to 502 (the Modbus server's port). The Modbus gateway is thus enabled.

If the serial port RTU device is the slave station, then the host computer Modbus TCP software connects to port 502 of Modbus gateway. At this time, Modbus gateway needs to work in TCP server mode. If the serial port RTU is the main station, the Modbus gateway works on the TCP client side, and the destination IP fills in the IP of the computer on which the Modbus TCP software resides, and the destination port is generally 502.

10.2 Storage Modbus Gateway

7146 is a Modbus gateway with register storage. Compared with ordinary ITTLAN5142 (the end of which is 2 or 0 is non-storage), 7146 can save the read contents of the registers in the gateway. In this way, the speed of Modbus TCP query can be greatly improved, and the performance is superior when supporting multi-host access.



As shown in figure 38: ordinary Modbus TCP data flow direction （1）-> （2）-> （3）-> （4）. In other words, Modbus TCP instruction is first converted into corresponding Modbus RTU instruction, and then the device responds Modbus RTU instruction to Modbus gateway, which is then converted into Modbus TCP again and sent to the monitoring host computer.

As we know, Modbus TCP is a network communication with a fast transmission speed, which can be answered within 3ms, while Modbus RTU is RS485 with a speed of only 9600bps. Generally, it takes at least 30ms to send and return an instruction. Such a common non-storage Modbus gateway has a relatively long query response time. In addition, if there are many upper computers to query data at the same time, the serial port will be congested. If the network is compared to the highway, the serial port is compared to the single-log bridge, then the original way is to pass the traffic flow of the highway on the single-log bridge.

Register save Modbus Gateway (7146) solves the above problem. It can temporarily save the register data obtained from the query in the Modbus gateway, so that when Modbus TCP query comes, Modbus gateway can immediately return the instruction, and truly give play to the rapid characteristics of Modbus TCP. On the other hand, 7146 can actively send instructions from the serial port to automatically update the data content of the currently saved register and save a copy of the latest register value.

In addition, 7146 is a fully automatic Modbus Gateway without configuration. Users do not need to configure required register addresses, function codes and slave station addresses. The 7146 will automatically recognize and dynamically add these registers according to Modbus TCP instructions sent from the network port.

When monitored by multiple computers, 7146 can show a good response speed. Regardless of the baud rate of the serial port, it can generally give the upper response data within 3ms. And it shows a good real-time update speed of serial data.

Modbus gateway with register storage is a real Modbus TCP to Modbus RTU, which gives full play to advantages of fast speed and multi-host simultaneous query of Modbus TCP.

Note that when the serial server is a TCP client, it does not have a storage function and will automatically switch to a non-storage type.

The features of the storage Modbus are listed below:

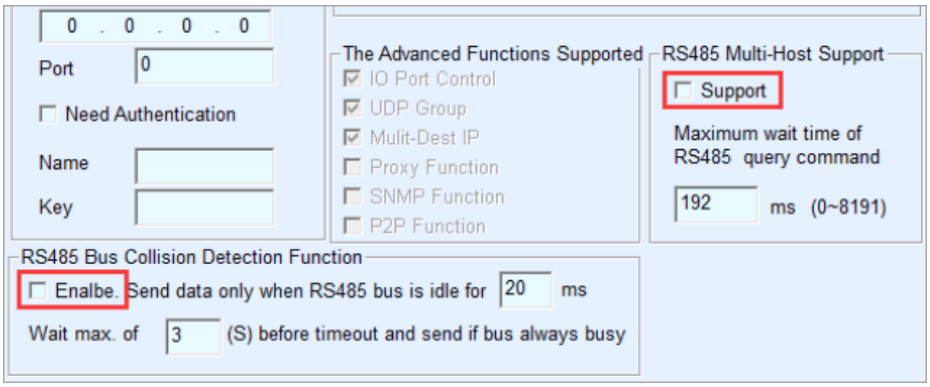
1. The first Modbus TCP query instruction is non-stored. Because you have to wait for the RTU device to slowly return data before returning the register contents to the network port.
2. If a particular instruction is no longer queried by the host computer on the network side within 5 seconds, the instruction is automatically deleted and no longer sent from the serial port to the RTU device.
3. Currently, it can store 10K Modbus cache. For ordinary single-register query, about 500 instructions can be stored at the same time.
4. When there are multiple instructions at the same time were queried, send in accordance with the order, send first instruction □ first instruction reply □ wait for 485 anti-collision time (refer to many host part) □ the next instruction to send... . Return to the first instruction after the last instruction has been answered.

10.3 Disable Storage Functionality

Although the stored Modbus has a fast response speed, some users do not want the RTU device to receive a large number of query instructions, which will affect the internal processing speed of the instrument. You can turn off the storage function at this point.

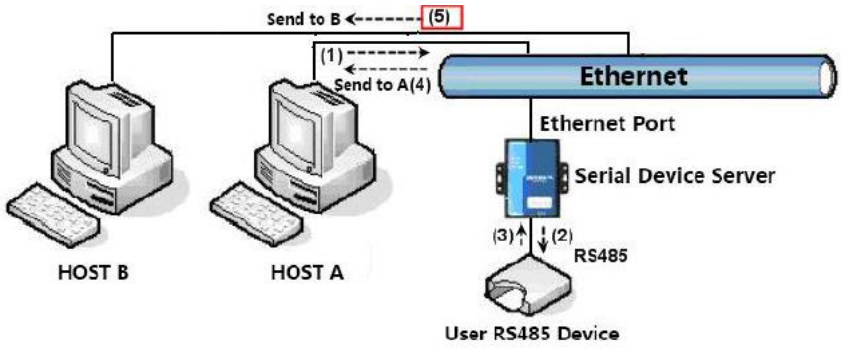
To disable storage, click the “more advanced options” button in the “Parameters Setting” dialog box and remove one of the support and one of the enable options shown in figure 34 by clicking ok. Then go back to device Settings and click modify Settings.

Note that when configuring the transfer protocol in a Web method, the default is a non-storage Modbus gateway.



10.4 Multi-host Function

As shown in figure 34, "RS458 multi-host support" and "RS485 bus collision detection function" are ITTLAN's multi-host functions. They are generally both enabled and disabled at the same time. After enabling, the device that converts the protocol into Modbus TCP has the function of storage Modbus gateway; otherwise, it is a non-storage Modbus gateway. If the conversion protocol is none, generally, the user-defined RS485 protocol can also have the function of serial devices accessed by multiple hosts at the same time, which cannot be achieved in the pure RS485 network, because the sending of multiple master stations at the same time will cause conflicts on the RS485 bus. The multi-host of ITTLAN serial port server can "coordinate" the RS485 bus so as to achieve the purpose of multi-host access.



As shown in figure 35, in normal mode, when two hosts, host A and host B, are connected to the serial port server at the same time, host A sends (1) instruction, RS485 device receives (2) instruction, and RS485 device returns (3) instruction, but the serial port server sends (4) to host A and (5) to host B at the same time. Since host B did not send a query, but it also received a reply instruction (5), host B may generate a communication exception error. In the multi-host mode, only instructions (4) will be given and no instructions (5) will be given, because the serial port server will automatically remember the host that needs to be returned and only return the instructions to the nearest communication host. The query of host A will only reply to A, and the query of host B will reply to host B.

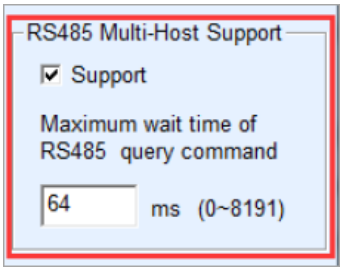
Another effect is that in normal mode, when host A and host B send data at the same time, the instruction combination will be generated on the RS485 bus, so that it cannot be recognized

normally. In the multi-host mode, the serial port server can schedule the priority of A and B on the use bus, so as to solve the conflict problem of simultaneous access of multiple machines in an effective manner.

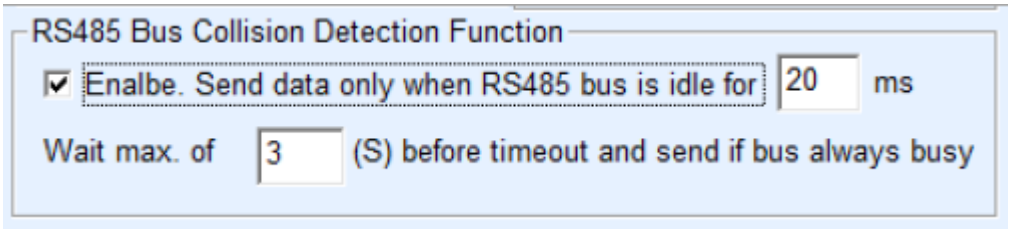
If the conversion protocol is "none", the default is not to enable multi-host function. If you need to enable multi-host, please click "more advanced options" in the device configuration dialog box, and then check "RS485 multi-host support".

10.5 Multi-host Parameter

The meanings of "RS458 multi-host support" and "RS485 bus collision detection function" are described as follows.



RS485 instruction reply timeout is: the maximum time interval between the start of sending this instruction and the receipt of the reply by the serial port server. The filling time should be greater than the actual maximum time interval. If the timeout is determined, the next instruction is sent.



RS485 bus collision time: indicates how many milliseconds the serial server waits to send the second instruction after receiving the reply of the first instruction. This parameter actually defines the speed of instruction rotation. This value is recommended above 20ms. The "maximum wait time is 3 seconds" parameter does not need to be modified.

When the user use ITTVirCom to select "Modbus TCP to RTU" as the transfer protocol, ZLVricom will automatically select the above two enabled boxes (unless the user manually enters the advanced option to remove), and the above two times will be automatically configured according to baud rate. However, if the user's Modus instruction is relatively long or the conversion protocol is "none", these two parameters need to be manually configured.

The following is the recommended setting values of the above parameters:

1. Figure 37 shows the "RS485 bus anti-collision time", which can be set as twice of the "packet interval" in the lower right corner of the parameter configuration interface, but the minimum value should not be less than 20.
2. Figure 36 shows "RS485 instruction reply timeout time", which is generally determined according to the length of the reply instruction. If the sending instruction is N bytes and the reply is M bytes, the recommended value is: "packet interval" * (N+M+5) +100.

10.6 Modbus Under Multi-destination IP

As shown in figure 40, if the serial port device (RTU device) is the master station and the Modbus TCP device (Modbus TCP device) is the slave station, and multiple network port slave station devices exist simultaneously. In this case, the serial port server can be used as the client to connect to multiple network port devices at the same time according to the method introduced in “7.2.2 Client Connect to Multiple Servers”.

At this point the need to achieve the function is: when instructions sent from the serial RTU can be sent to more than one network device, the network device identify whether to send their own through the Slave ID field, only the network devices corresponding to Slave ID make a response. After the network port reply is sent to the serial port server, it is converted into RTU instruction and output from the serial port to the RTU device.

At this point, it is important to note that the two checkmarks shown in figure 37 as "RS485 bus anti-collision time" and figure 41 as "RS485 instruction reply timeout time" need to be removed. Otherwise, the above forwarding function cannot be realized.

Another application method is that although the serial port server connects multiple network port devices as a Client, the RTU device is not the main station, but the network port device sends first and the RTU device answers (as the slave station). At this time, "RS485 bus conflict prevention time" and "RS485 instruction response timeout time" need to be checked, so that multiple hosts can access a RTU device at the same time.

11. Registration Pack and Heartbeat Pack

Registration and heartbeat packs are a feature that suitable for communication between devices and cloud-based software.

11.1 Registration Packet

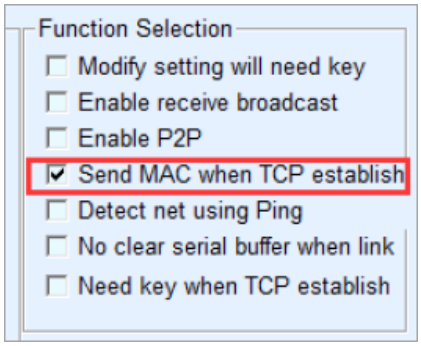
The registry is defined as a string of code that is sent to the software when a TCP connection is established between the computer software and the serial server module (hereinafter referred to as the module) so that the software knows which module is communicating with it. This string of codes is the registry.

The registration package is very suitable for the monitoring of the Internet of things, because the cloud software generally runs on the public network server of the Internet, while the modules are scattered in various collection and monitoring points. How to make the cloud software identification module is very important and necessary for the realization of Internet of things communication.

Ittelecom serial server to provide the following types of registration.

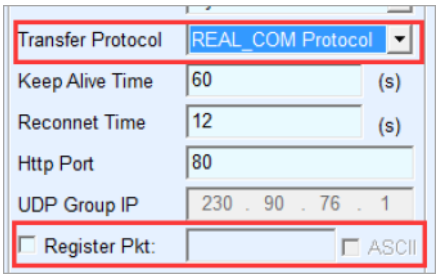
11.1.1 Connect to send MAC Addresses

Connect to send MAC addresses: this is not only for model 4 (like 5143), but also for regular models. It does this by sending its MAC address to the cloud when the module is connected to the cloud. Since the MAC address is unique, the device can be uniquely identified. This approach is simple and effective without having to write a registry for each device. To use it, click "more advanced options" in the device Settings dialog box, find "send MAC address when TCP is established" in the middle and upper part, check the box in the front, then go back to the Settings interface and click "modify Settings".



11.1.2 Realcom Agreement

The Realcom protocol is a mature protocol with registry and heartbeat packages that users can use to implement registry and heartbeat package functionality. The way to enable Realcom protocol is to select "REAL_COM protocol" as "REAL_COM protocol" in the device Settings dialog box.



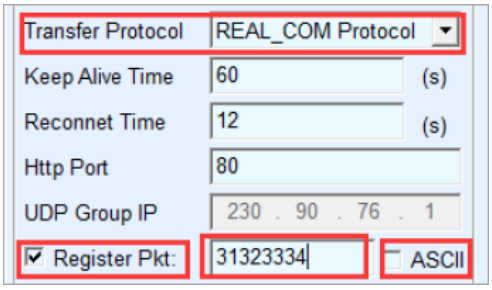
It is not a transparent transport after the Realcom protocol enabled. It has the following features:

1. When a TCP connection is established between the device and the cloud, the device automatically sends a hex registry FA 07 13 02 FA 02 MAC[5] MAC[4] MAC[3] MAC[2] MAC[1] MAC[0] FA FF. The MAC[5]~MAC[0] is the MAC address of the device.
2. When the device sends data to the network, it will automatically add the 3-byte header prefix of FA 01 01.
3. The device sends a 00 byte heartbeat packet to the software every keep-alive time.

The REAL_COM protocol can be used as a registry for devices because the registry contains MAC addresses. But because it is a fixed format, it can only be designed in the cloud software that REALCOM protocol to be compatible with this way.

11.1.3 Custom Registry

The custom registry format allows users to fill in an arbitrary registry format. The method is: in the device setting interface, the configuration is as follows:



The difference with the REAL_COM protocol is that the registry is enabled and the registry information 31 32 33 34 is filled in. Note that this is hexadecimal, which means that the actual data sent is the string 1234. If you need a string display, click the "ASCII" option.

When the device is connected to the cloud software, it can automatically send the hexadecimal registration package of 31 32 33 34. This registration method is more flexible, can let the device to adapt to the existing cloud registration format; However, there is no wildcard like MAC in the registry, so it is tedious to configure a different registry for each device. The above two methods

of sending MAC addresses and REALCOM are the same for each device configuration, but they are naturally different for different MAC registries.

The maximum registered package length is 33 bytes. This mode supports UDP mode registration packages and heartbeat packages.

Note that if the wifi parameter or multi-destination IP parameter is set, it will conflict with the custom registry heartbeat packet. At this time, it is recommended to use the transcode configuration file described in the next section to realize the registry and heartbeat packet.

11.1.4 Transcode Configuration Files

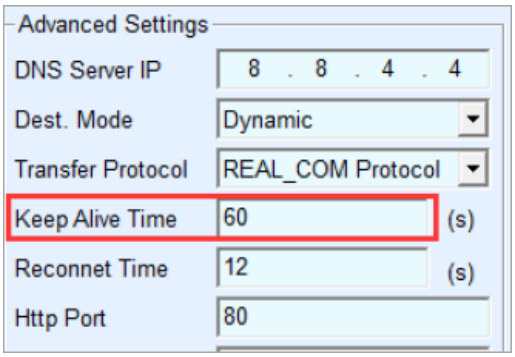
ITTLAN 5143 series supports the "transcoding" function, which can write a transcoding configuration table for the serial server, so as to realize the user's fully customized registration package, and can use MAC address wildcard, can solve the trouble of writing a custom registration package for each device, and there is no limit on the length of the registration package.

Specific use method can consult ITTLAN about "transcoding" function configuration, or refer to “12 Transcoding Function”.

11.2 The Heartbeat Packets

Heartbeat packet is mainly used to detect whether the communication link is disconnected. The implementation method is that the device sends a heartbeat packet data to the server software every once in a while, which will be discarded by the server after receiving and will not be regarded as valid data for communication.

Heartbeat packet has two main functions: first, it can let the upper computer software know that the device is in the active state; Secondly, if the device fails to send a heartbeat, the device on the TCP client will automatically re-establish a TCP connection, so it is a means to restore network communication.



As shown in figure 49, the sending time of the heartbeat packet is set by the "Keep alive time".

11.2.1 Implied Heartbeat

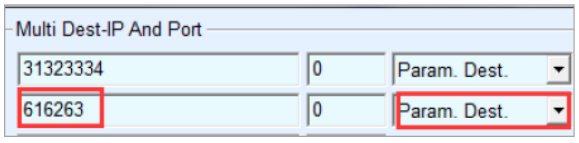
Even if no heartbeat packets are set, the ITTLAN device enables the implied heartbeat when it is on the TCP Client. Therefore, the implied heartbeat function means that the device has sent data, but the server cannot receive the heartbeat data. Therefore, it cannot play the first function of heartbeat packet, that is, the server can detect whether the device is active or not. However, since the device actually sends data, it can play the second function of heartbeat packet, that is, the function of the device to detect whether the TCP connection is normal. Once the detection is broken, the TCP connection can still be re-established automatically.

11.2.2 REALCOM agreement

As described in the “8.1.2 Realcom Protocol”, the Realcom protocol can send a byte data of 00 at every guaranteed time, which is the heartbeat packet of the Realcom protocol.

11.2.3 Customize Heartbeat Packets

First, fill in the registration package according to the method of “8.1.3 Custom Registration Package”. Then increase the heartbeat packet as follows: click the "more advanced options" button in the device Settings, write the hexadecimal heartbeat packet in the second line of the multi-destination IP and port, and change the option on the right to "Parameter Packet Destination".



Note that the total number of registered and heartbeat packets is less than 33 bytes. The first line is actually the registry.

12. HTTPD Client Communication Function

This function is used to send the data from the serial port server directly to the server program based on the web architecture, which can simplify the software development workload in the cloud.

When the collection terminal of the Internet of things interacts with the web server (HTTPD program), if the data can be submitted to the web server in the standard format of HTTP GET and POST instructions, the web server can use the existing PHP /asp language to process and store the data. This saves the user the effort of redeveloping the web application interface.

To support this feature, download a configuration file of httpd.txt from the ITTLAN serial port server. Download can be done using the firmware upgrade feature of ITTVirCom.

The features of ITTLAN HTTPD Client communication include:

1. Device send: support to directly convert serial data into HTTP format by GET/POST, which can be directly recognized by the server.
2. Web server issue: the Web server can also send the required data to the serial port server through the GET/POST instruction, where the valid data content can be output from the serial port of the serial port server. When the serial server receives the data, it can also send a specific reply to the Web server indicating that the data has been received.
3. Support arbitrary conversion of input and output data between hexadecimal and string, convenient for the Web server to send data in character mode, and the serial port to control the serial device in hexadecimal data output.

For more information, please refer to the "ITTLAN HTTPD Client Communication Mode" document.

13. Transcoding Functions

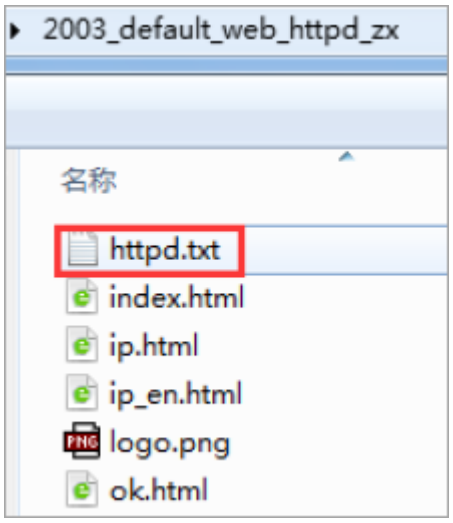
ITTLAN serial port server transcoding function can achieve different device protocol into a unified protocol function.

As shown in table 5, when the network port receives 01 02 03 04 the serial port will output instructions of a1 a2 a3 a4. Similarly, when the serial port receives b1 b2 b3 b4, the network port actually sends 11 12 13 14. This enables the conversion of different protocols. This is just a

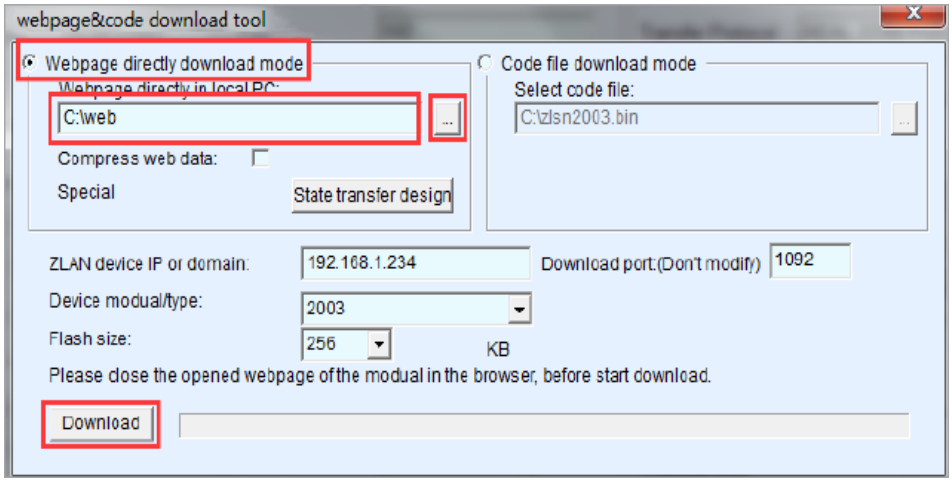
simple example of a "transcode" function that can actually be converted to more complex protocols.

13.1 Enable the Transcoding

The configuration files converted by the command are written to the httpd.txt file and then downloaded to the ITTLAN serial port server. To do this, create a new httpd.txt document in the web page directory. If the user does not have a web directory, make a new webs directory and then make a separate httpd.txt file.



Now click the "upgrade firmware" button in "device configuration", and the web and firmware download page will pop up:



As shown above, select the web directory to download, then select the web directory you just created, and then click the "download" button to download httpd.txt into the device. Above IP, model, space size, port will be automatically filled in without configuration. Note: any time httpd.txt is updated, the device needs to be repowered.

13.2 Case of Transcoding Implementation

Because transcoding function is more complex, the length is longer. Without going into detail here, it is also possible to write the configuration file httpd.txt by using

ITTLAN engineer if necessary. ITTLAN can also provide the configuration file httpd.txt for the existing case. At present, there are several typical cases as follows:

1. Fixed command translation and conversion.
2. Complex command conversion: the conversion of commands with wildcards, which can be shifted and assembled, such as the conversion of non-standard RS485 instructions into standard Modbus RTU instructions.
3. Multi-TCP connection recognition: when there are multiple TCP connections, the network port send data to the serial port will increase 4 byte prefix of IP+ port. When a serial port is connected to a network port, the IP address + port can also be added to send data to the specified TCP connection.
4. Add the MAC address before sending TCP or UDP to identify the device.
5. IP filtering: the device only receives data from the network port with the specified IP address.
6. IO pin control: control the level of some output pins by any specified command.
7. Automatically query the content of the instrument of the lower computer, assemble the returned instruction and send it to the cloud software in a fixed format.
8. Implement heartbeat packets and registration packets with wildcards of any length.
9. Multiple serial instructions are combined and sent to the cloud after the combination.
10. The super-long serial instruction is split into several smaller instructions to be read by the network device.

13.3 Notice

1. When “httpd.txt” is not stored, the device will be used as a normal serial port server. Transcoding is disabled.
2. To delete “httpd.txt” you can simply change “httpd.txt” to “1.txt” and then download it again. Re-downloading the firmware of the device will also delete the “httpd.txt” file, and downloading the firmware progress bar will delete the “httpd.txt” file if it only moves 1 frame. Refer to the following sections for download firmware methods.
3. The device must be restarted after downloading the “httpd.txt” file.
4. When an “http.txt” editing error causes syntax parsing to fail, which can cause a device startup exception, set the “def” switch (reset switch) of the device to “On” and re-power

it. This will start without “httpd.txt”. Please download the new “httpd.txt” or delete “httpd.txt” and then set “def” back to the “Off” position.

14. Network Interface to Modify Parameters

Network interface modification parameter is a function to search and modify device parameters like ITTVirCom software, that is, to manage the device and modify parameters through the network interface of the serial port server. Suitable for users who integrate search and configuration functions into the user's software.

Network interface modification parameters are realized through "UDP management port protocol", such as:

1. Computer software sends UDP broadcast packets with destination port of 1092 in the network. When the device receives the data packet, it will return its information to the computer software to achieve the purpose of searching the device.
2. The computer software sends the UDP modification parameter command to the 1092 port of the device to achieve the purpose of modifying the device parameter.

The detailed introduction of network interface modification parameters can be referred to the document of "UDP Management Port Protocol of ITTLAN Networking Product". It can also be implemented directly using the device management function library of the “14 Device Management Function Library”.

15. Device Management Function Library

This feature is suitable for users who need to integrate device management functions into their own software.

The described "UDP management port protocol" is integrated into the device management function library ZLDevManage. This is a Windows platform development library of DLL, can be called by VC, VB, Delphi and other development tools.

Provide detailed API interface introduction documents and VC call Demo case. Can realize the device search, parameter modification, P2P function call and so on.

Can be gained from ITTLAN website development libraries: search for "equipment management function library" in http://ITTELECOM.CO/download pages. Please refer to "ITTLAN WinP2p and equipment management development library" for details.

16. Serial Port Modify Parameters

Users can read and set parameters by sending instructions to the serial port of the serial port server. Suitable for users choose chip or module level products to control and configure through the serial port. Parameters that can be set include: IP address, baud rate, device name, working mode, etc. After setting the new parameters, you can restart the serial port server through the serial port instruction.

ITTLAN serial instruction has the following characteristics:

1. The serial instruction uses 10 bytes of data leading code, so there is no need to distinguish the communication data or command through the pull down and pull up of another configuration pin, and there is no need to switch the command mode and communication mode, which is more flexible and convenient to use.
2. The command set contains a variety of command formats such as save parameters, do not save parameters, restart devices, and so on.
3. Can achieve a variety of applications, such as reading the MAC address of the serial port server, such as changing the working mode of the serial port server when switching from TCP Server to TCP Client mode, can actively connect to the server; You can disconnect from the TCP server when switching from the TCP Client to the TCP Server.

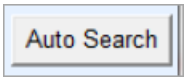
Please refer to “Serial port modification parameters and hardware TCPIP protocol stack” for detailed operation methods.

17. Remote device management

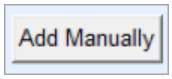
The so-called remote device management refers to the maintenance and management of the device through ITTVirCom software, including restarting the device, modifying parameters and upgrading firmware. This feature is suitable for users who manage devices through ITTVirCom**.**

For ITTVirCom software, the device can be managed remotely as long as it can be found in the device list. Remote management of equipment can be divided into the following situations:

1. Automatic search: under the same switch of the device and computer, whether in the same network segment or not, the method of ITTVirCom on the computer search for devices are: ITTVirComsend broadcast query，All devices will reply their parameters to the ITTVirCom tool after receiving the query. This method searches all devices at once.

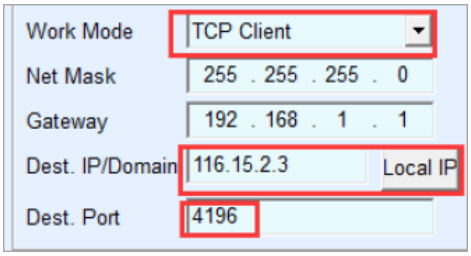


2. Manual add: divided into two cases:

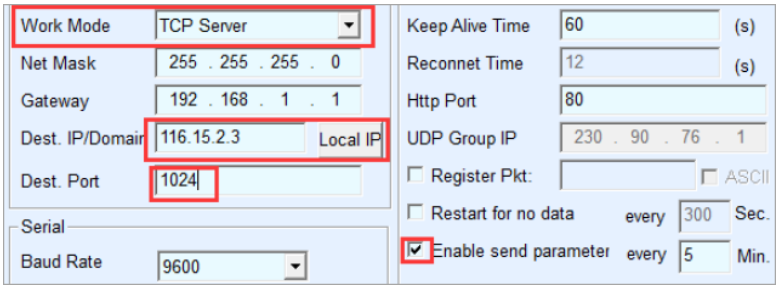


a. Large routers split the network: in some large networks, the broadcast packets are split by the router so that the broadcast packets cannot reach the device, but pinging device IP is enabled. At this point, you generally need to manually add to solve. The method of manual addition is to click "manual addition" in the "device management" dialog box to add the first and last IP to query the devices one by one.
b. Public network server query internal network equipment: the serial server in the internal network and as a TCP Server mode, ITTVirCom in the public network IP server. At this time, a port mapping of 1092 UDP should be made on the router of the network where the device is located, which is mapped to the IP of the device. Then, ITTVirCom will manually add the device, and the IP is the public network IP of the device.

3. TCP Client: when the device is a TCP Client, it initiates a TCP connection to port 4196 of the destination IP (116.15.2.3). When the connection is established, each hold time automatically sends its own parameter system to the UDP port (not TCP) on the destination port (4196 in this case,), enabling ITTVirCom to search for devices on this computer (116.15.2.3). If the destination port is not 4196, the default parameter receiving port of ITTVirCom needs to be modified by changing the menu/configuration/software configuration/default listening port, and then starting ITTVirCom will ignore the execution if TCP port conflict pops up.



4. Timing sending parameters: even if the serial server is in the mode of TCP server, you can check the "timing sending parameters" function and send parameters to the destination port of the destination IP (in this case, 116.15.2.3) every 5 minutes. ITTVirCom, which receives parameters on this port of the server, can manage these devices.

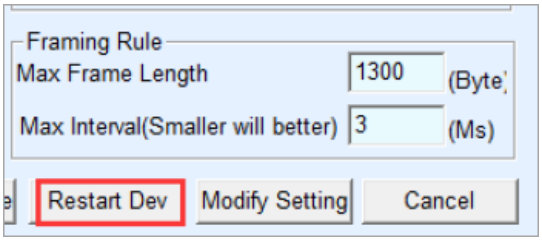


To facilitate device identification, give the device an easy-to-remember name if remote administration is required.

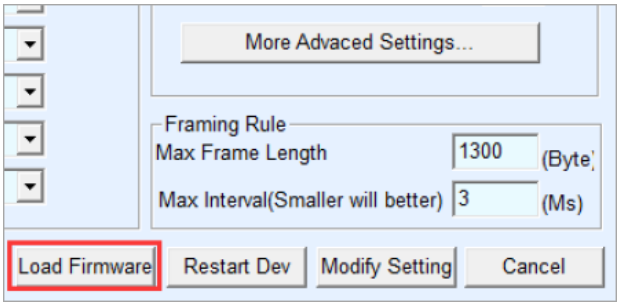
18. Firmware upgrade method

ZLSN7146 can upgrade their own programs, but not each other. Firmware updates can be applied to devices found in the device list, whether through automatic search, manual addition, or P2P search.

1. Get the ZLSN7146 firmware file from ITTLAN, such as 1.539(7146).bin.
2. In the ITTVirCom tool, first search for the device that needs to be upgraded, and then go to the device parameter edit dialog box. First click "restart device" once.



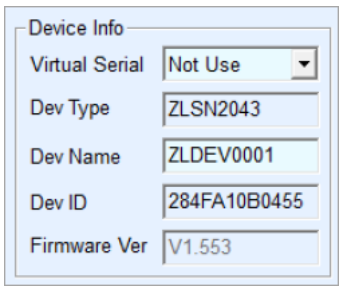
After the device is restarted, use the same method to search the device again and enter the dialog again. Click the "upgrade firmware" button in the lower right corner of the dialog box.



3. As shown, select the “program file download” radio option. In the program file, select the firmware file. The IP address part of the serial port server has been filled in automatically, no need to write again, and the module type/model has been selected automatically. Then click download.

4. The download bar starts to move, and the download takes about 30 seconds. The ACT lights on the device will flash during the download, and the LINK lights will flash a few times at the end of the download. Then the program pops up the prompt box of "do not power off the LINK light blinking device after transmission". Note: this is only after the transmission is completed, it will take about 3 seconds to write the flash process. At this time, the LINK light will flash, please don’t power off in the time.
5. Generally after the completion of the download the procedures will automatically restart, generally without power. See the running indicator light flashing. If there is no automatic restart, please power on again after the LINK light flashing stop for more than 30 seconds.
6. Web configuration interface update: after firmware upgrade, the configuration page inside the module also needs to be updated, otherwise it cannot be configured through the Web again, but it does not affect the communication. Web pages can also be downloaded without a web configuration. To download the Web, change the "program file" download mode to "Web directory download" as shown in the figure. And select the root directory where the local web page is located as the directory where the web files need to be downloaded (the directory can be obtained from ITTLAN), click download, and download all the files in the local web page directory to the internal file system of the device.

7. **Note:**
   a. If download under AP mode
   b. If the prompt fails, the device will not be damaged. Please restart the download. In addition, at the end of the download, please do not power off when the LINK light flashes, otherwise the device will be damaged.
   c. Check out the firmware version number of ITTVirCom to find out whether the new firmware has been downloaded successfully.



After-service and technical support

Web: http://www.ittelecom.co

Email: support@ittelecom.co