

# Frequently Asked Questions (FAQ) About Modbus Conversion

## Modbus Protocol Conversion between Serial and Ethernet Devices

Modbus protocol is widely used in many automation applications and many devices support serial based Modbus RTU protocol. These devices generally support the RS-232 or RS-485 interfaces, often using a DB-9 connector or terminal block. Modbus RTU devices are very easy to implement and troubleshoot at a very low cost, which explains why Modbus RTU has become so popular. But as more and more industrial applications continue to migrate to Ethernet for more complex networking needs, many systems now use Modbus TCP on a SCADA host in the control room, resulting in communication issues between Modbus RTU and Modbus TCP protocols. The following FAQs and answers seek to help you identify and prevent common Modbus protocol conversion issues.

The following FAQs include:

1. To connect Modbus RTU serial devices to an Ethernet network, do I need a specific protocol conversion gateway? Or can a serial device server achieve the same result?
2. If I have more than one Modbus RTU device connected to different serial ports on a gateway, how do I plan the TCP connection architecture? Can I use a single connection, or do I need to use a separate connection for each serial port?
3. How do I use a gateway to allow multiple SCADA hosts to access the same Modbus RTU devices simultaneously?
4. I have two working Modbus master devices (PLC or HMI). How do I exchange the data between these two systems?
5. I have multiple Modbus RTU devices I need to query. I can use multiple Modbus commands to read all register data but it takes too much time. Is it possible for the gateway to get the data actively and put them into single register so I can just use one Modbus command to get all the data?

# ▲ Modbus Gateway FAQs

## Question 1:

To connect Modbus RTU serial devices to an Ethernet network, do I need a specific protocol conversion gateway? Or can a serial device server achieve the same result?

### Answer:

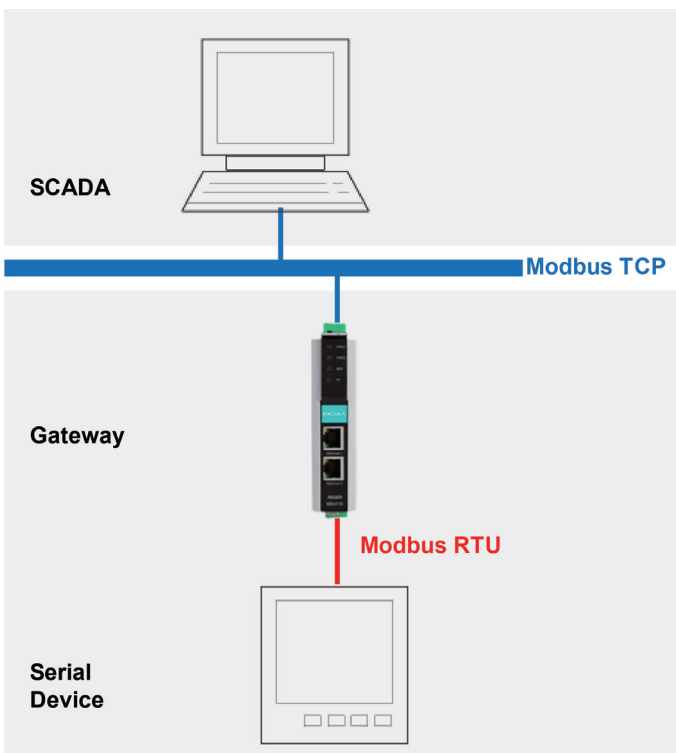
Before answering this question, you have to know which Modbus driver on the SCADA host you want to use. There are four possible options:

- (1) SCADA host with Modbus TCP driver,
- (2) SCADA host with Modbus RTU driver – with built-in serial port,
- (3) SCADA host with Modbus RTU driver – without built-in serial port,
- (4) SCADA host with “Ethernet Encapsulation” driver

### (1) SCADA host with Modbus TCP driver

Use a protocol conversion gateway for this. With the gateway’s help, you can use Modbus TCP protocol to communicate with Modbus RTU devices.

There are many “Modbus gateway” solutions in the automation market that enable Modbus TCP connection for Modbus TCP slave devices. When the gateway receives the Modbus TCP request, it converts the packet to a Modbus RTU packet and sends it to the Modbus RTU devices immediately.

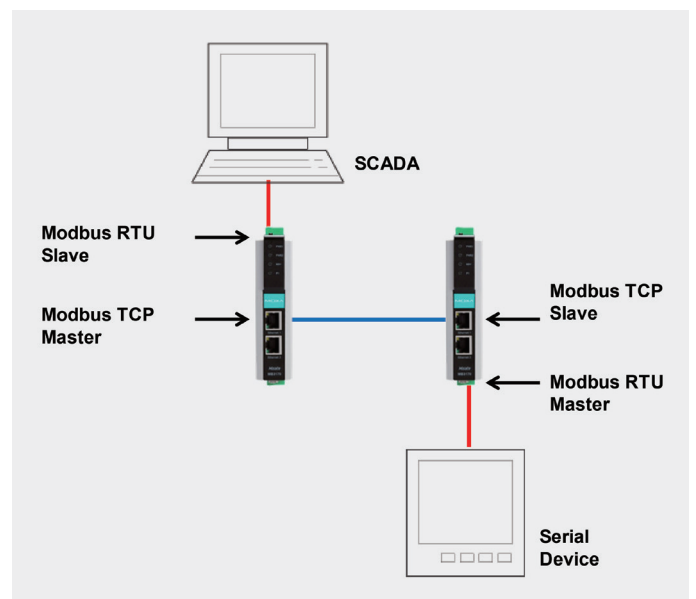


### (2) SCADA host with Modbus RTU driver–with built-in serial port

Choose this option if you just want to connect your existing SCADA host and Modbus RTU devices to an Ethernet network. If your original SCADA host has a built-in serial port, you can use a pair of gateways to solve this issue.

As shown in the topology pictured, the gateway can convert the Modbus RTU packet into a Modbus TCP packet and back to a Modbus RTU packet.

However, if you don’t have a built-in serial port on your SCADA host, this is not the right solution for you. Instead, consider option 3.

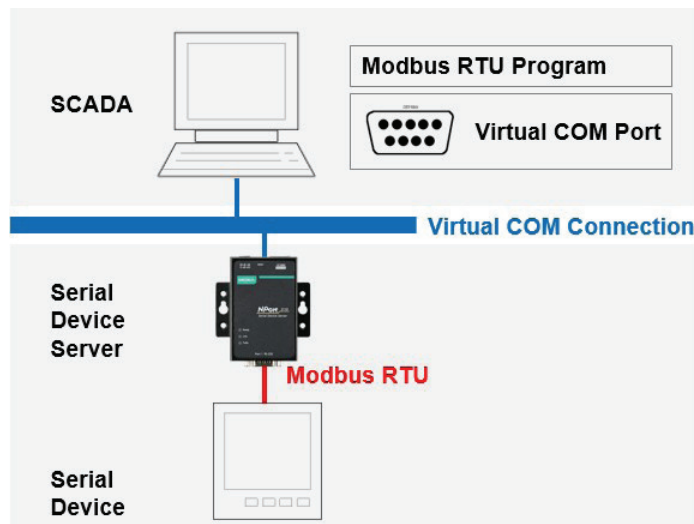


## ▲ Modbus Gateway FAQs

### (3) SCADA host with Modbus RTU driver—without built-in serial port

If you want to use your existing SCADA program and devices, but your original SCADA host does not have a built-in serial port, then use a “serial device server” to build a “virtual COM port” for the serial port on the remote serial device server connected to your serial devices. This configuration allows you to access the remote serial devices through the serial device server as if it had a native COM port.

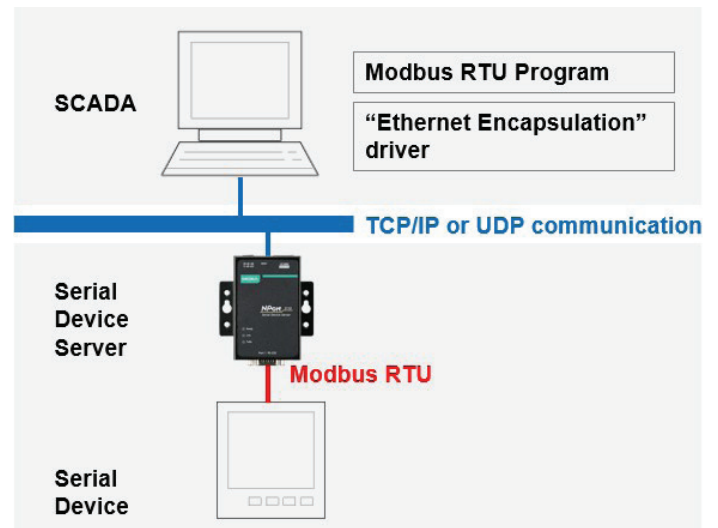
The serial device server will install the “virtual COM port driver” on your SCADA host to create a “virtual COM port”. To enable the virtual COM port, the serial device server must be configured to “virtual COM” mode. All data sent to this virtual COM port will be transferred to the remote serial port of the serial device server. All actions for the modem signal will also be handled in the same way. Since you can use the virtual COM port in the same way as a native local COM port, you can send the Modbus RTU request to the COM port directly, just as you would if it were a physical COM port.



### (4) SCADA host with “Ethernet Encapsulation” driver

If you do not have a built-in serial port on your SCADA host but not want to install a “virtual COM port” driver, you can consider an “Ethernet encapsulation” driver instead. However, your SCADA software must be able to support the “Ethernet Encapsulation” driver, which isn’t always the case. Generally speaking, “Ethernet encapsulation” drivers are best if you have in-depth serial and TCP/IP knowledge.

For this option, you will need a “serial device server” with “raw socket mode” or “Tunneling mode” selected, which means the connection between the host and serial device server uses transparent TCP/IP or UDP communication without any protocol when SCADA sends Modbus RTU packets to field devices. The serial device server needs to be configured properly because Modbus RTU uses interval timeouts to determine the end of the packet. If the Modbus RTU packet is divided into two or more TCP/IP or UDP packets, you may encounter problems. If you are not confident in properly handling packet transfers between serial lines and Ethernet networks, you’re probably better off with the gateway solution (option 2) or the virtual COM driver solution (option 3).



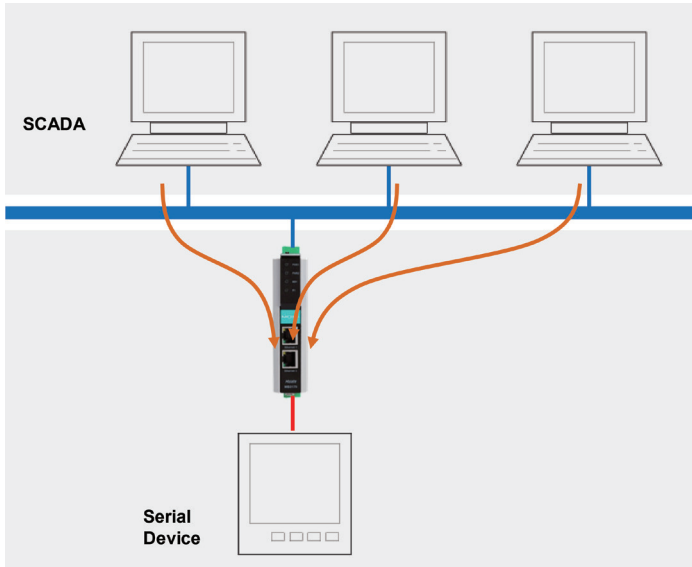
Although serial device servers can also connect Modbus RTU devices to an Ethernet network, the gateway solution (option 2) can meet almost every system requirement. Your host must be able to support Modbus TCP connections, this shouldn’t be a problem since Modbus TCP is very popular and already widely supported. Here are a few situations where you will need to use a designated gateway solution:

#### 1. Multi-master or redundancy.

Ethernet connections not only provide the benefit of remote access, but also multi-connection access capability. Most gateways can support up to 32 connections, which means up to 32 SCADA hosts can query the Modbus RTU devices simultaneously. Although it will be difficult for a serial device server to

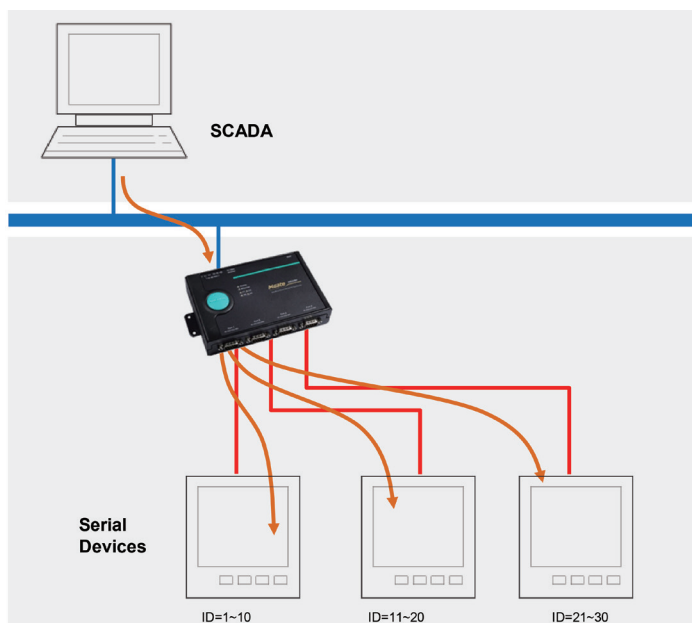
# ▲ Modbus Gateway FAQs

provide network redundancy in this situation, as most serial device servers do not support multiple masters, a gateway on the other hand will have no problem.



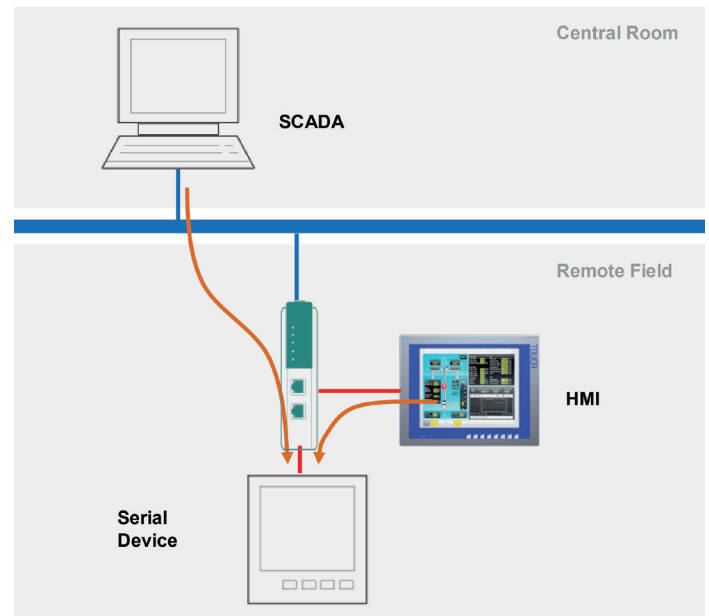
## 2. Single connection for multiple Modbus RTU devices

Sometimes, you may want to use a single connection on a SCADA host to query multiple Modbus RTU devices connected to different serial ports. In this case, a gateway is the only solution that can process the routing mechanism. Multiple serial port gateways can be used and configured to send the Modbus request to the correct serial port based on unique slave IDs. A serial device sever can't handle such a complicated requirement.



## 3. Simultaneous device access for old Modbus RTU HMI and new Modbus TCP SCADA

Although Ethernet connections offer easy-to-deploy remote access, there may be times you want to keep your existing local HMI connections. The problem is the serial port on the device is already connected to the gateway so there is no available serial port for the HMI connection. In this situation, some gateways provide a “serial redirector” to overcome this obstacle. The “serial redirector” is very similar to a router in that the gateway can transfer the request between different serial ports based on the slave ID.



## Conclusion

There are many different solutions for serial-to-Ethernet communications. Although these solutions can be as simple as a transparent data transfer between serial and Ethernet ports, a designated gateway is generally a far better option if you need to handle industrial protocols such as Modbus. A designated gateway may involve a larger initial investment, but it provides more stable communications in the long run and can recognize Modbus packets to properly handle entire packets.

# ▲ Modbus Gateway FAQs

## Question 2:

If I have more than one Modbus RTU device connected to different serial ports on a gateway, how do I plan the TCP connection architecture? Can I use a single connection, or do I need to use a separate connection for each serial port?

## Answer:

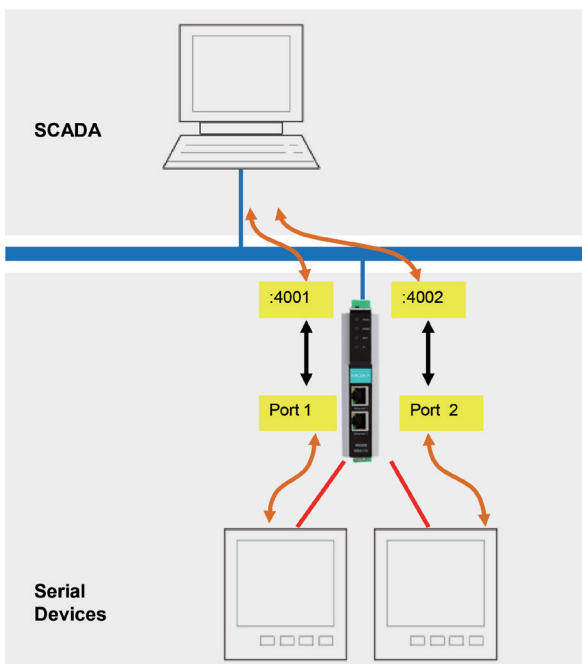
Most gateways provide flexible solutions that can be used to plan TCP connection architecture for more than one Modbus RTU device connected to different serial ports of the gateway. In general, there are three different methods you can use that are based on the routing mechanism:

- (1) Link serial port to unique TCP port,
- (2) Link serial port to unique IP address, and
- (3) Use routing table.

### 1) Link serial port to unique TCP port

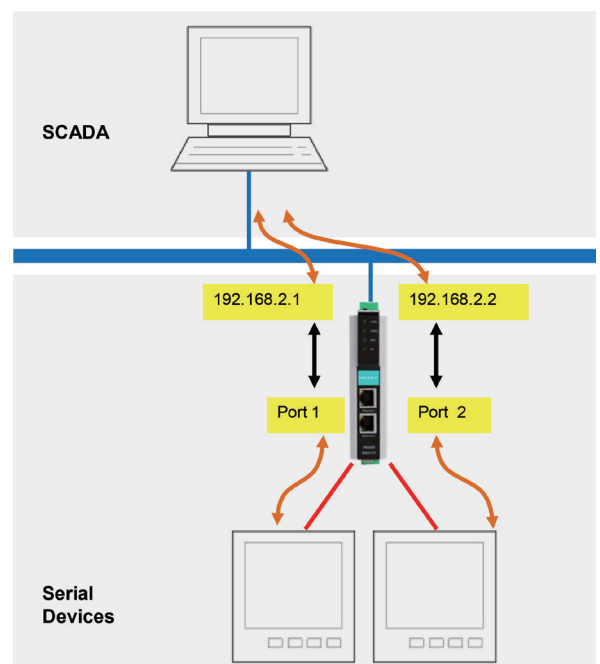
This is the most popular method for planning your gateway topology. In the gateway configurations, each serial port will be identified by one unique TCP port. For example, 4001 for serial port 1, 4002 for serial port 2, etc. If you want to communicate with Modbus RTU devices on serial port 1, you need to build a Modbus TCP connection to 4001. The gateway will transfer the Modbus packet between TCP connection on TCP port 4001 and serial port 1.

In this topology, the SCADA driver needs to build multiple Modbus TCP connections.



### (2) Link serial port to unique IP address

This is very similar to option 1, except the gateway uses different IP addresses to identify different serial ports. For example, 192.168.2.1 for serial port 1, 192.168.2.2 for serial port 2, etc. If you want to communicate with Modbus RTU devices on serial port 1, you need to build a Modbus TCP connection to 192.168.2.1 with TCP port 502. The gateway will transfer the Modbus packet by TCP connection between 192.168.2.1:502 and serial port 1. In this topology, the SCADA driver also needs to build multiple Modbus TCP connections. Although this topology requires multiple IP addresses, some Modbus TCP clients only support TCP port 502. If that is the case, option 1 may not be viable so may need to choose this option.



## ▲ Modbus Gateway FAQs

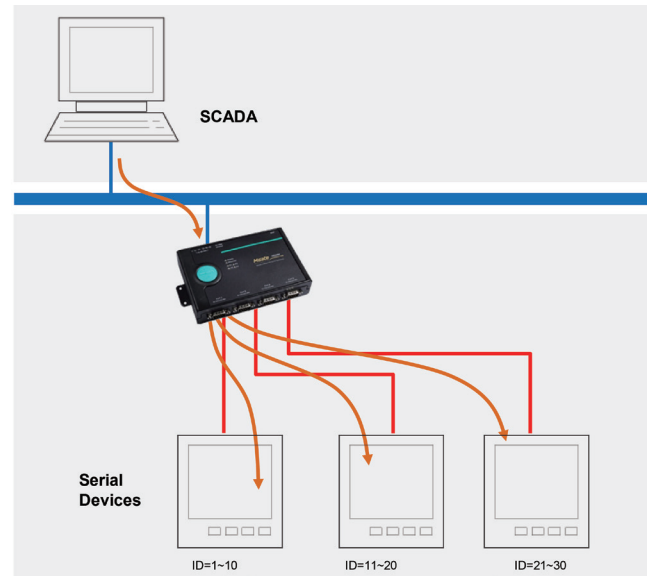
### (3) Use routing table

This topology uses signal connection to communicate with multiple devices. To make sure the request can be transferred to the correct serial port, the gateway and the routing path both need to be configured properly. For example, serial port 1 handles all Modbus packets that have slave IDs 1 to 10, serial port 2 handles slave IDs 11 to 20, etc.

Because this topology only uses one connection, it will be slower than options 1 and 2. But if you have constrained by budget concerns and the performance is still acceptable for your needs, then a single connection can be a good option.

#### **Note:**

If you connect more devices to a single serial port, or if you link multiple serial ports to one TCP connection, you will increase the Modbus polling time. But again, if you want to increase the polling speed, more TCP connections might be required and you need to confirm this will not increase your SCADA costs.



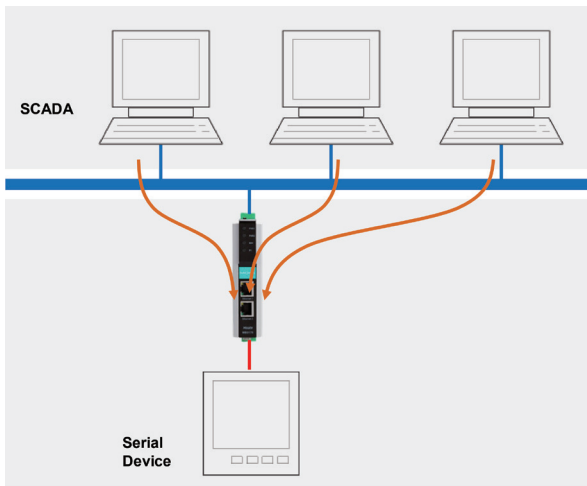
## ▲ Modbus Gateway FAQs

### Question 3:

How do I use a gateway to allow multiple SCADA hosts to access the same Modbus RTU devices simultaneously?

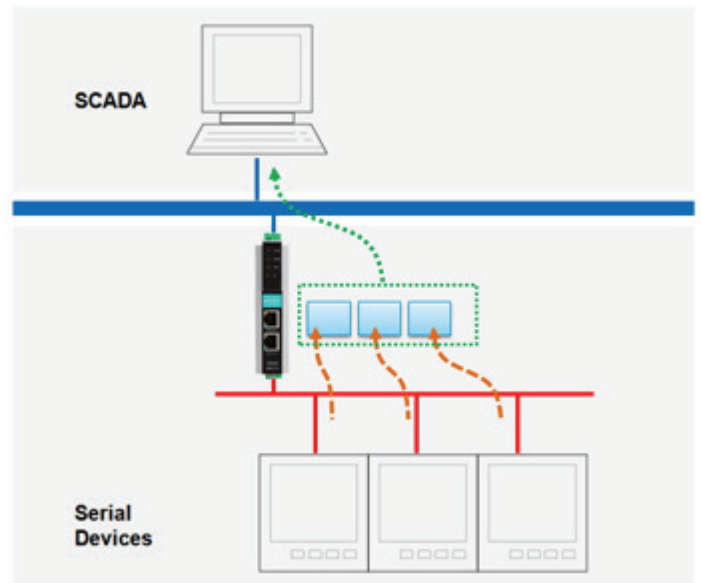
### Answer:

Although the gateway can handle this properly, remember that the serial port bandwidth stays the same. So if multiple requests are received through the same serial port, you will likely experience a delay because the gateway needs to handle the earlier requests in the queue first. That means you need to calculate the proper polling time if you want to enable multiple masters to access the same Modbus RTU device simultaneously.



For example, if one request takes 100 ms, five multiple connections would require you to wait at least  $100 \text{ ms} \times (5-1) = 400 \text{ ms}$  before sending the next request. That means each SCADA host will need to use 400 ms (plus some tolerance) for polling cycle time.

But the above is not the only solution. Some gateways support an “agent” mode, which actively and continuously retrieves data from connected devices. The updated data is stored in internal memory, which is used to reply to host requests. Although this solution is faster and more efficient, the data retrieved might not be the most current in real time.





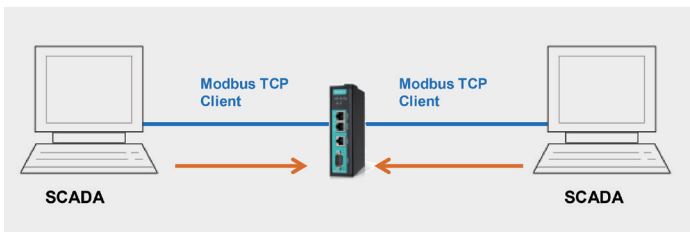
## ▲ Modbus Gateway FAQs

### Question 4:

I have two working Modbus master devices (PLC or HMI). How do I exchange the data between these two systems?

#### Answer:

In order to exchange data between two Modbus master devices, you need a gateway that can support master-master mode. In this mode, the gateway will work as the slave for both sides. One master can write the data to the internal memory of gateway, and another master can get the data to achieve the data exchange. Depending on the gateway you use, you can have both sides support Modbus RTU and Modbus TCP, or just Modbus RTU on one side and Modbus TCP on the other.

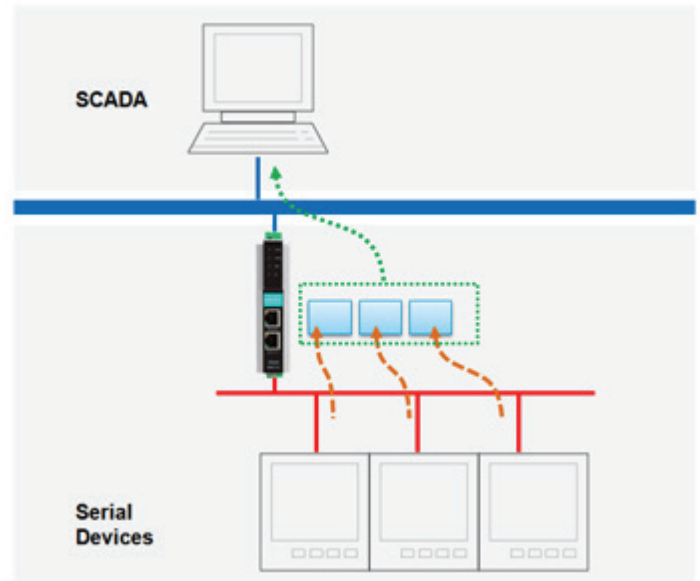


### Question 5:

I have multiple Modbus RTU devices I need to query. I can use multiple Modbus commands to read all the register data but it takes too much time. Is it possible for the gateway to retrieve the data actively and put them into single register so I can just use one Modbus command to retrieve all the data?

#### Answer:

In order for the gateway to actively retrieve data from multiple Modbus RTU devices and place them into a single register, you need an agent gateway to store the data in its internal memory. The gateway also needs to send a Modbus command to each Modbus RTU device one by one. All the data will be arranged into a single data block on the gateway's internal memory for you to retrieve with just one Modbus read command.





# ▲ Modbus Gateway FAQs

## Glossary

### Serial Device Server

A “serial device server” is a standalone device that has at least one Ethernet port and one or more serial ports. Serial device servers are equipped with an embedded network operating system and allow computers to access serial devices over a network. A serial device server can transfer data transparently between serial interface and Ethernet interface, converting all serial data received for Ethernet protocol communications, as well as converting all Ethernet data received for serial protocol communications.

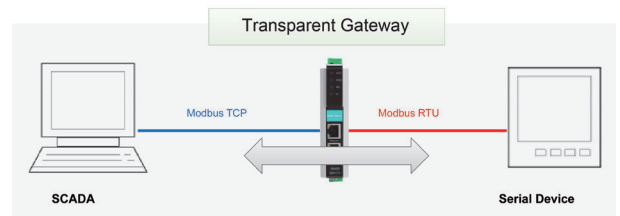
Modern serial device servers can also provide a “virtual COM port” for host computers that lack an extra serial port, essentially converting an Ethernet port to a COM port. Besides these basic functions, more sophisticated serial device servers can even support PPP in serial or Telnet protocols over Ethernet networks. A serial device server can even be used for console management (which is why some vendors may call it a “console server”) or remote terminal applications in old banking systems (which is why it also sometimes called a “terminal server”).

### Virtual COM Port, Virtual COM Port Driver

A “virtual COM port” is not a “real” (i.e., physical) COM port on the host computer. Instead, a “virtual COM port driver” program under Windows is installed on the host computer to completely emulate the behavior of a local COM port. The driver is used for ports on a serial device server that connects to the host via a TCP/IP network. The serial port on the remote serial device server will exhibit the same behavior as a local COM port. This “virtual COM port” is configured to link to a serial device server. For example, COM3 is serial port 1 on a remote serial device server that has an IP address of 192.168.2.1. So when you write data to this port, all data will be sent to “192.168.2.1@serial port 1” by “virtual COM port driver”. All the requests to this virtual COM port will be re-directed to “192.168.2.1@serial port 1”. Because newer host computers often lack built-in serial ports, “virtual COM ports” are an invaluable tool for connecting legacy equipment in industrial automation.

### Transparent Gateway

A “transparent gateway” is the most basic way to use a Modbus gateway. Since Modbus RTU and Modbus TCP have the same PDU (Protocol Data Unit), and the only difference is the address field, it is easy for a gateway to transfer data between Modbus RTU and Modbus TCP. So when the gateway receives the Modbus TCP packet from the Ethernet network, it can simply replace the address field to fit Modbus RTU addressing and immediately send the packet to the serial port. When the gateway receives a reply from the Modbus RTU devices, it will reply to the Modbus TCP client.



### Agent Gateway

An “agent gateway” provides another way to use a Modbus gateway to transfer the data between Modbus TCP and Modbus RTU. The agent gateway has its own internal memory to store data temporarily, and it will query the data from the connected devices continuously. When the SCADA driver query is received, the gateway will use the data stored on its internal memory to reply to the query. In this way, the gateway works as an agent to query device data actively. This feature can be used for protocol conversion where:

- (1) The two protocols use different packet structure. For example, PROFIBUS and Modbus, EtherNet/IP, PROFINET, etc.
- (2) The two protocols use different cycle times. Some protocols—such as PROFIBUS, PROFINET, and EtherNet/IP—exchange data within very short time cycles, which “transparent gateways” cannot accommodate.

