COMPACT INDUSTRIAL RS-485 DEVICE SERVER

(RS-485 TO ETHERNET / MODBUS RTU TO MODBUS TCP CONVERTER)

CommFront
Revision: 1.0

USER MANUAL

Model: DVS-485-2

Contents

1.0 INTRODUCTION

The DVS-485-2 is a compact, rugged, industrial-grade RS-485 device server that supports one channel bi-directional RS-485 to Ethernet conversion which can be used to transmit serial data over Ethernet and TCP/IP network. The applications of the DVS-485-2 include acting as an RS-485 to Ethernet converter or a Modbus RTU to Modbus TCP protocol converter; controlling RS-485 serial devices from remote Modbus TCP software or Modbus RTU master via Ethernet network; making a wide variety of M2M communication applications possible by enabling client access independent of any operating system or proprietary protocol; extending RS-485 serial communication distance over Ethernet network; and converting conventional serial data network to TCP/IP network. The DVS-485-2 was designed for harsh industrial environments and withstands a wide temperature range of -40ºF to 185ºF (-40ºC to 85ºC) and is equipped with 15kV ESD and 600W surge protection. The unit supports TCP and UDP for both server and client modes, and it features static IP and DHCP for minimizing configuration errors caused by manual entries. The DVS-485-2 features auto-negotiation for 10 or 100M data rates, and it also supports MDI and MDI-X (auto-crossover); no DIP switch or jumper settings are required.

1.1 Features

- Compact, rugged, industrial-grade RS-485 device server
- RS-485 to Ethernet converter
- Modbus RTU to Modbus TCP protocol converter
- Modbus gateway for master and slaves
- Designed for harsh industrial environments
- Direct 35mm DIN-Rail mounting
- Easy configuration via AT commands (RS-485) or built-in web server
- Supports static IP and DHCP
- Supports TCP and UDP for both server and client modes
- Supports Httpd client mode
- Supports up to 32 nos. of RS-485 devices (daisy-chain)
- Supports serial baud rates from 600 bps to 460 Kbps; RFC2217 compliant
- Supports up to 8 simultaneous client connections
- Supports auto-connection (TCP client mode) and auto-restart for smooth operation
- Supports short-connection, index, and other advanced features
- Supports MAC address and user-defined registry packets
- Supports heartbeat packets for maintaining effective network connections
- Auto-negotiation for 10/100M data rates; MDI and MDI-X auto-crossover
- 15kV ESD and 600W surge protection
- Operating temperature: -40ºF to 185ºF (-40ºC to 85ºC)
- Compliance: CE, FCC
- Surface Mount Technology manufactured to RoHS and ISO-9001 standards
- 5-year manufacturer’s warranty

1.2 Factory Settings

IP Settings:	Static IP
IP Address:	192.168.1.201
Subnet Mask:	255.255.255.0
Gateway:	192.168.1.254
DNS:	8.8.8.8
COM Port Settings:	115.2Kbps, 8, N, 1
Local Port:	502
Remote IP:	192.168.1.211
Remote Port:	20000
Modbus:	Modbus RTU <-> Modbus TCP (OFF)
User Name:	admin
Password:	superuser

Note: To restore factory settings, press the “reset” button for 5 to 15 seconds.

1.3 Hardware

1.3.1 Specifications

Compatibility:	EIA/TIA RS-485; IEEE802.3
Power Source:	5 to 36 VDC
Power Consumption:	Less than 1W
Configuration:	AT commands (RS-485) or built-in web server
Ethernet Network:	10/100M Auto-Negotiation; MDI-X auto-crossover
Network Protocols:	IPv4; TCP/UDP/HTTP; ARP; ICMP
Network Buffer Size:	TX: 6KB; RX: 4KB
IP Settings:	Static IP or DHCP
Operating Modes:	TCP Server & Client; UDP Server & Client; Httpd Client
Max. No. of Client Connections:	8
Serial Port:	600 bps to 460 Kbps; RFC 2217 compliant
Serial Port Buffer Size:	RX: 2KB
Communication Protocols:	RS-485; Modbus RTU; Modbus TCP
No. of RS-485 nodes:	32
Protection:	Up to 15kV ESD; 600W Surge Protection
Distance (Ethernet):	328ft (100m)
Distance (RS-485):	4000ft (1.2km)
Connectors (Ethernet):	1x RJ45
Connectors (RS-485):	1x 3-Way Terminal Block (A, B, GND)
Connectors (Power):	2-way terminal block
Dimensions (L x W x H):	4.3x1.1x2.6 in (110x28x65 mm)
Weight:	0.2 lb (90 g)
Operating Temperature:	-40ºF to 185ºF (-40ºC to 85ºC)
Operating Humidity:	5% to 95% Non-condensing

1.3.2 Connections

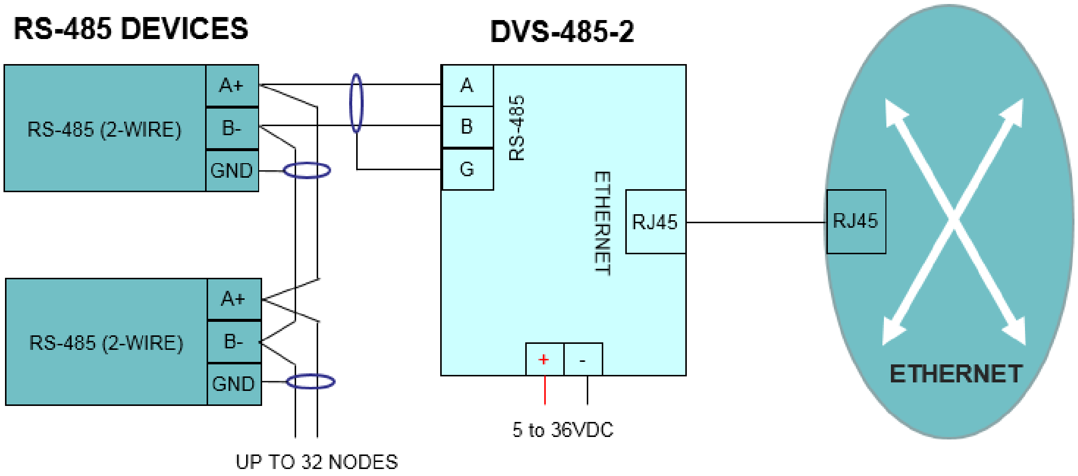
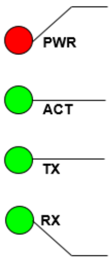
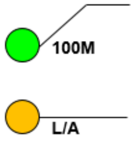


FIGURE 1: DVS-485-2 CONNECTION DIAGRAM

1.3.3 LED Indicators



PWR Indicator
ON: Power supply is on
OFF: Power supply is off
ACT Indicator
Flashing: The unit is in working condition
OFF: The unit is not working
TX Indicator
Flashing: Serial port is transmitting data
OFF: Serial port is idle
RX Indicator
Flashing: Serial port is receiving data
OFF: Serial port is idle



10/100M Indicator
ON: The unit is connected to a 100M network
OFF: The unit is connected to a 10M network or is disconnected
Link/Act Indicator
Flashing: Sending/receiving data
OFF: No network connection

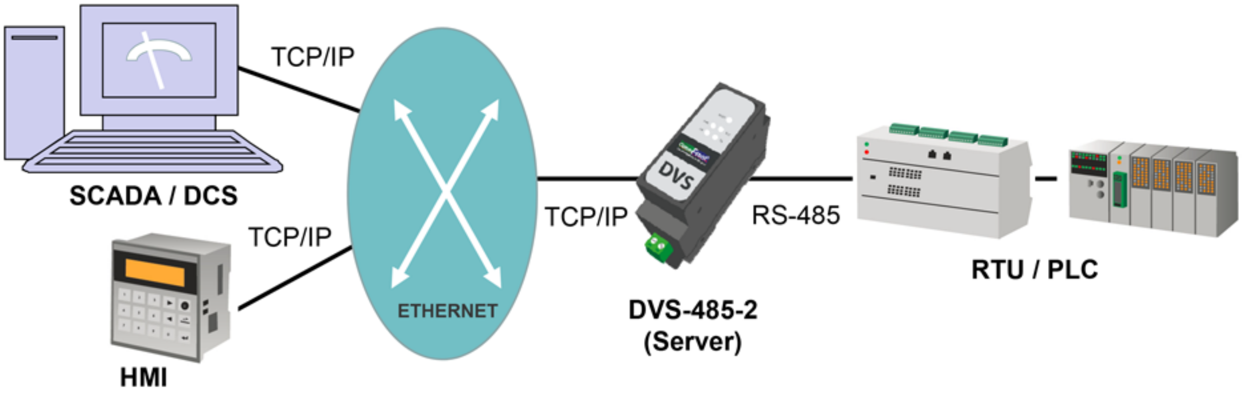
1.4 Typical Applications

CommFront device servers are designed for harsh industrial applications and provide a simple yet reliable solution for the following applications:

- M2M
- IIoT
- Data Acquisition
- Mass Transit
- Point of Sale
- Traffic Signals
- Security Systems
- Energy Plants
- Water and Sewage
- Medical Applications
- Factory/Industrial/Building Automation

1.4.1 Device Server

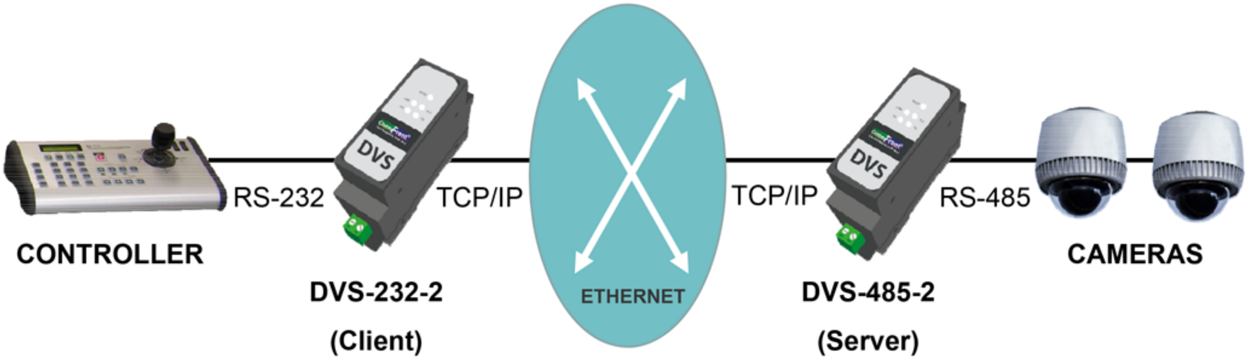
The device server makes a wide variety of M2M communication applications possible by enabling client access independent of any operating system or proprietary protocol. In addition, because conventional serial protocol are converted into TCP/IP protocol, the device server can be accessed by remote TCP/IP software and/or control stations. This makes the remote control, management, monitoring, and diagnose possible and, in turn, adds an unprecedented level of intelligence and efficiency to the operation.



APPLICATION 1: DEVICE SERVER

1.4.2 Serial over Ethernet

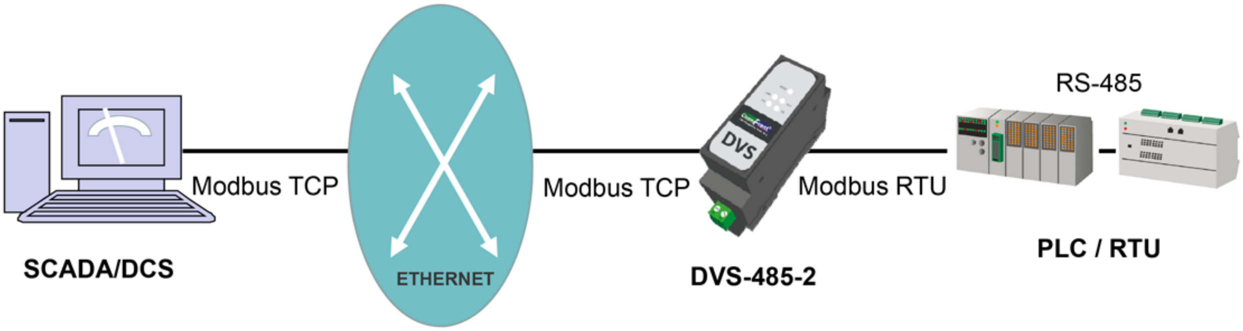
The DVS-485-2 supports one channel bi-directional RS-485 to Ethernet conversion which can be used to transmit serial data over Ethernet and TCP/IP network. “Serial over Ethernet” allows you to make use of an existing Ethernet network and extend the distance across a facility or to other facilities all over the globe.



APPLICATION 2: SERIAL OVER ETHERNET

1.4.3 Modbus Gateway

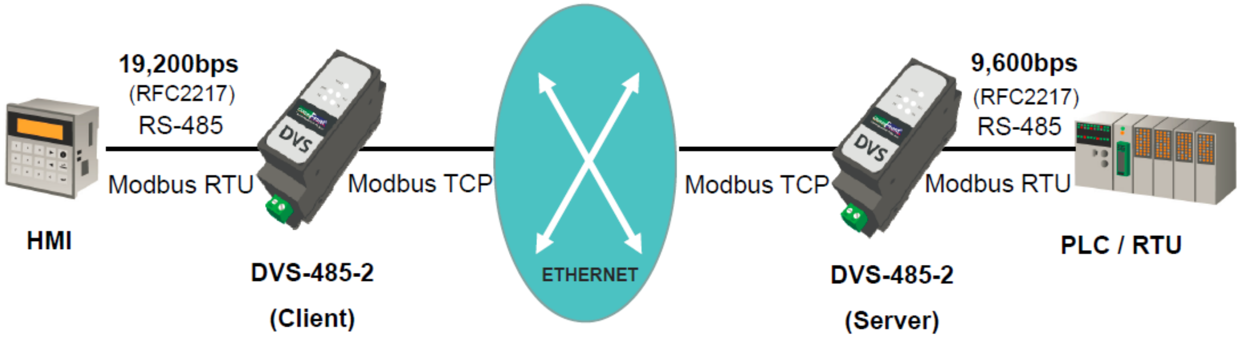
Modbus RTU is the most widely used industrial protocol. A typical Modbus system consists of one master and multiple slaves that run over the same single twisted pair cable. The DVS-485-2 can act as a Modbus gateway that encapsulates Modbus packets in standard TCP/IP packets. This enables the filed Modbus devices to connect and communicate over existing Ethernet and fiber networks instantly. In addition, conventional Modbus RTU has a limitation of 247 nodes per network, while Modbus TCP networks can have as many slaves as the physical layer can handle, which often go around 1,024.



APPLICATION 3: MODBUS GATEWAY

1.4.4 Modbus Protocol Converter

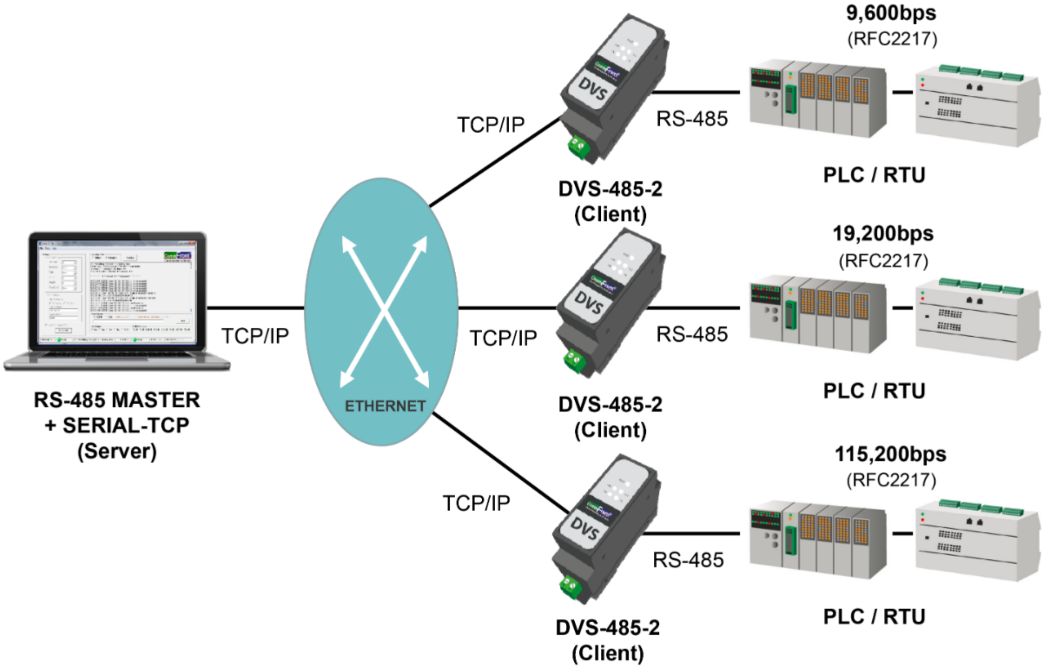
The DVS-485-2 can act as a Modbus protocol converter that encapsulates Modbus packets in standard TCP/IP packets. A Modbus protocol converter allows conventional Modbus RTU master and slave devices that run on twisted-pair cable to run over Ethernet and fiber networks with greater distance and flexibility (e.g. the COM port settings for client and server can be different).



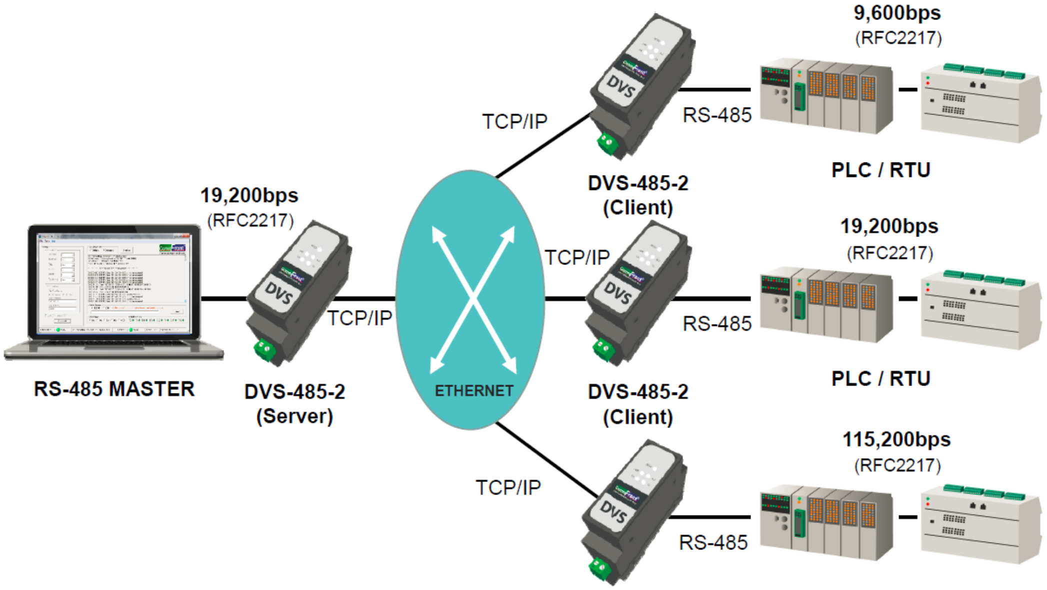
APPLICATION 4: MODBUS PROTOCOL CONVERTER

1.4.5 Converting RS-485 Network into TCP/IP

By using device servers, you can convert existing RS-232/RS-485 field devices such as PLC and RTU into TCP/IP devices, which can directly be controlled by the TCP/IP control software. With CommFront's Serial-TCP software (an advanced serial to TCP/IP bridging software), you can also convert conventional serial control software into TCP/IP software instantly and seamlessly without a need of software modifications. In addition, various serial devices with different COM settings (RFC2217 supported) can be merged into the same TCP/IP network, thus making the entire network conversion fast, easy, and secure.



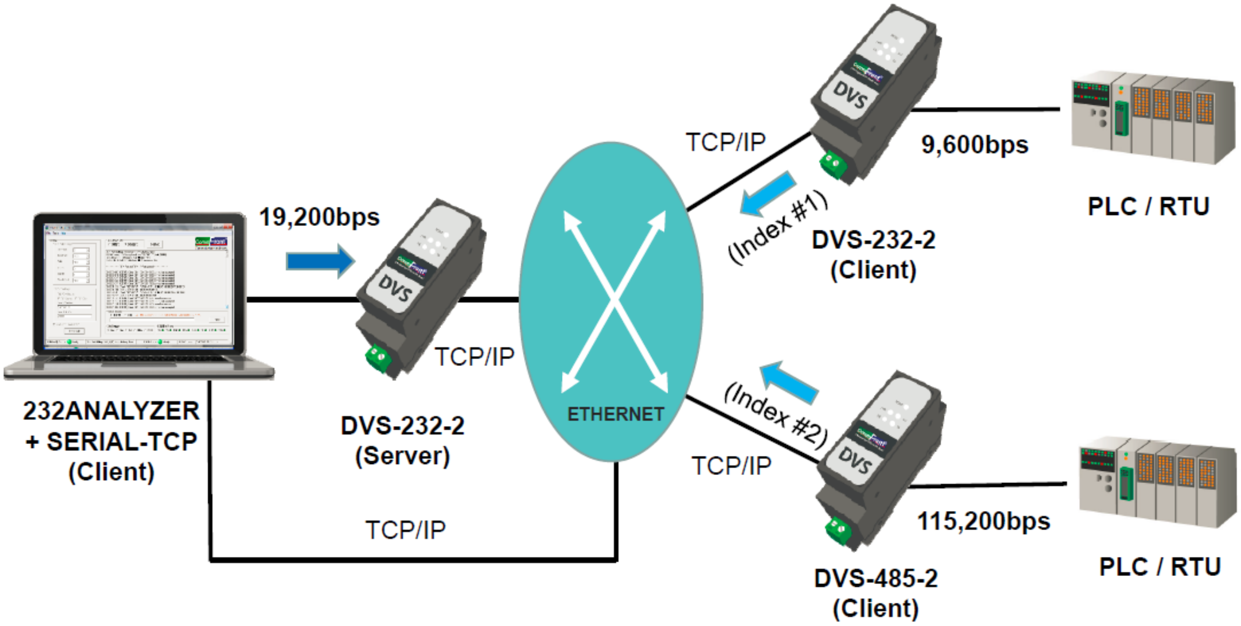
APPLICATION 5-1: CONVERTING RS-485 NETWORK INTO TCP/IP (SERIAL-TCP)



APPLICATION 5-2: CONVERTING RS-485 NETWORK INTO TCP/IP (DVS-485-2)

1.4.6 Remote Control and Monitor

From award-winning 232Analyzer Protocol Analyzer to Advanced Serial to TCP/IP Bridging Software, and from industrial Serial Converters, Isolators, Repeaters, and Hubs to the compact Device Servers, CommFront offers a complete hardware and software solution for controlling and monitoring conventional serial and modern TCP/IP communications. In addition, not only you can test, debug, control, and monitor serial and TCP/IP communications including Modbus RTU/ASCII and Modbus TCP, but you can also integrate our industry-proven software and hardware products with your mission-critical projects without a need of software or hardware modifications.



1) Via COM port: In TCP server mode, the device server can turn on the "Index" feature so to see the data source on the 232Analyzer software (see "Index" feature for more information).
2) Via TCP/IP: When working together with the Serial-TCP software via a pair of virtual COM ports, the conventional serial control software such as the 232Analyzer software can be turned into a TCP/IP control software instantly and seamlessly without a need of software or hardware modification. In addition, with the Serial-TCP software's built-in "debugging" feature, all data between serial port and TCP/UDP port can be monitored and logged.

APPLICATION 6: REMOTE CONTROL AND MONITOR

2.0 SETTING UP THE DEVICE SERVER

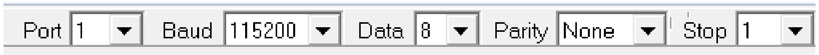
The DVS-485-2 offers two easy ways for configuration: AT commands via COM port or Web Management via the built-in web server (see "Web Management").

2.1 AT Commands (optional)

Assume that your PC is at the same sub network of "172.16.0.x".
Note: Skip this step and go directly to "Web Management" if your PC is at the same sub network of "192.168.1.x".

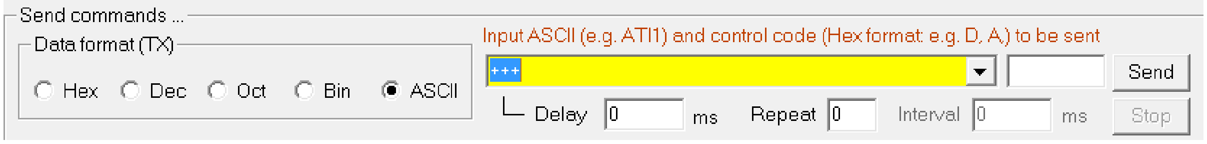
1) Download and install the free version of CommFront's 232Analyzer software (download link: https://www.commfront.com/pages/downloads).

2) Run the 232Analyzer software and set COM port settings to 115.2Kbps, 8, N, 1. Then select "ASCII" for both TX and RX data formats.



Port 1 Baud 115200 Data 8 Parity None Stop 1

3) To enter AT Command mode, first send the ASCII code: **+++**



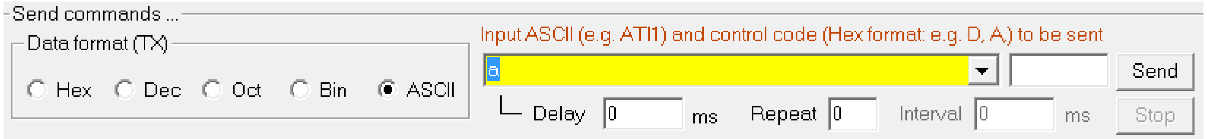
Send commands ... Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent: +++

Delay 0 ms Repeat 0 Interval 0 ms

4) The DVS-485-2 responds with an ASCII code: **a**

5) Send the second ASCII code within 3 seconds: **a**



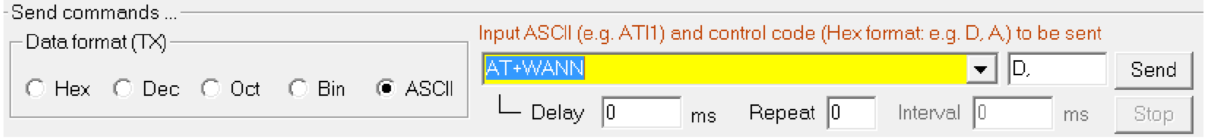
Send commands ... Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent: a

Delay 0 ms Repeat 0 Interval 0 ms

6) The DVS-485-2 responds with an ASCII code and it now enters into "AT Command" mode: **+ok**

7) Query current network settings: **AT+WANN<CR>**
Note: <CR> equals to "D" in Hex code, remember to enter "D," into the second box below



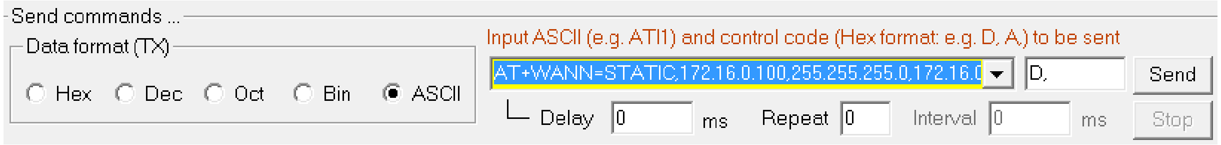
Send commands ... Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent: AT+WANN D,

Delay 0 ms Repeat 0 Interval 0 ms

8) The DVS-485-2 responds with the current network settings:
<CR><LF>+OK=STATIC,192.168.1.201,255.255.255.0,192.168.1.254<CR><LF>

9) Change network settings to the same sub network as your PC, e.g.: **AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0.1 <CR>**



○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent

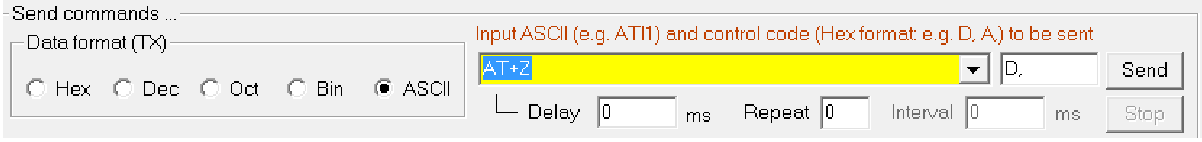
AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0

D,

Delay 0 ms Repeat 0 Interval 0 ms

10) The DVS-485-2 responds with a confirmation code: **+ok**

11) Restart the DVS-485-2 for the new settings to take effect: **AT+Z<CR>**



○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent

AT+Z

D,

Delay 0 ms Repeat 0 Interval 0 ms

12) The DVS-485-2 is now at the same sub network of your PC. Proceed to web management for further configurations (see "Web Management").

Note: For more detailed information about AT Commands, please refer to Appendix: AT Commands Set.

2.2 Web Management

Web management offers a user-friendly interface for fast and easy configuration. All parameters and functions can be set from the web management tools. You can go directly to web management tools if your PC is at the same sub network of "192.168.1.x". Consult with your network administrator on how to change your PC's network settings, if necessary.

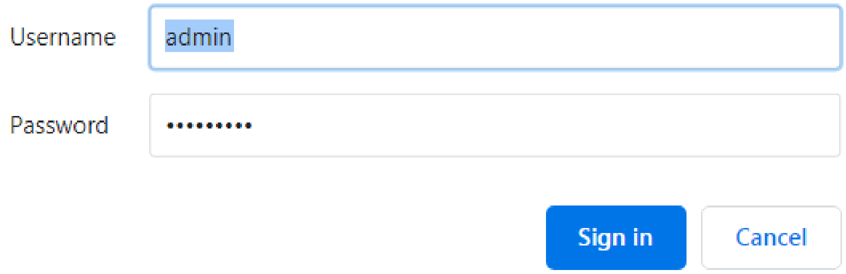
3.0 WEB MANAGEMENT

Web management offers a user-friendly interface for fast and easy configuration. All parameters and functions can be set from web management. The default IP address of web management is: 192.168.1.201. You can go directly to web management if your PC is at the same sub network of "192.168.1.x". Consult with your network administrator on how to change your PC's network settings, if necessary.

3.1 Logging On

A user name and password is required to log onto web interface.

- default user name: admin
- default password: superuser



Username: admin

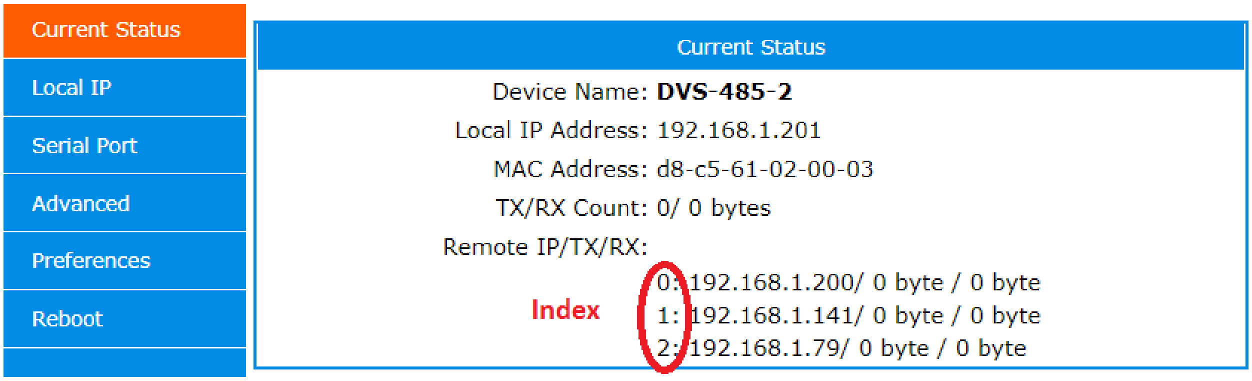
Password: ••••••••••

Note: User name and password can be changed under the "Preferences" web page.

3.2 Current Status

The "Current Status" web management page gives you a brief view of the current status of the device including MAC address, the current IP address, and the current remote connections as well as the number of bytes sent and received.

Note: For server mode, the number of remote TCP connections are indexed and can be used for identifying the source of TCP packets (see "Index" feature).



Current Status

Device Name: **DVS-485-2**

Local IP Address: 192.168.1.201

MAC Address: d8-c5-61-02-00-03

TX/RX Count: 0/ 0 bytes

Remote IP/TX/RX:

0: 192.168.1.200/ 0 byte / 0 byte

1: 192.168.1.141/ 0 byte / 0 byte

2: 192.168.1.79/ 0 byte / 0 byte

3.3 Local IP Settings

The "Local IP" web management page allows you to set the following parameters:

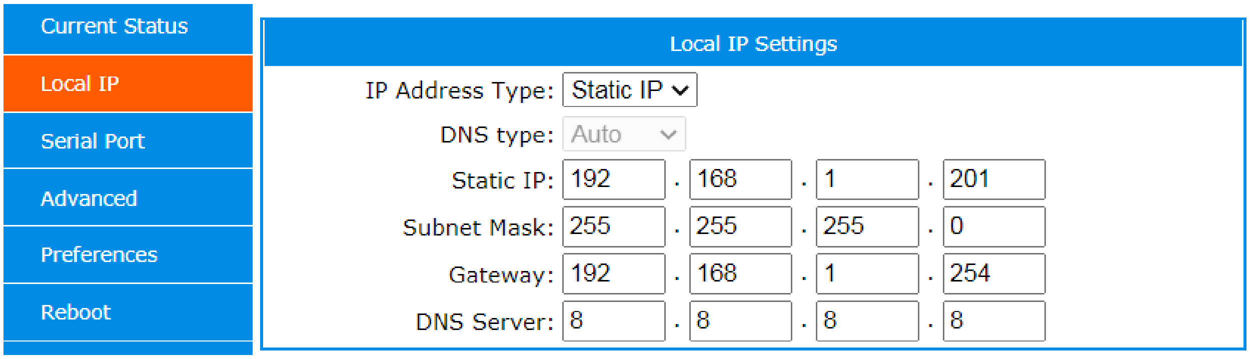
- IP Address Type: Static IP or DHCP
- DNS type (for DHCP): Auto or Manual
- Static IP Address
- Subnet Mask
- Gateway
- DNS Server

Table 1: Default Local IP settings

IP Settings:	Static IP
DNS Type (for DHCP):	Auto
IP Address:	192.168.1.201
Subnet Mask:	255.255.255.0
Gateway:	192.168.1.254
DNS Server	8.8.8.8

Notes:

- Remote IP Address and Port Mapping are to be set at "Serial Port" web page.
- Please consult with your network administrator for IP and Port settings, if necessary.



Local IP Settings

IP Address Type: Static IP

DNS type: Auto

Static IP: 192 . 168 . 1 . 201

Subnet Mask: 255 . 255 . 255 . 0

Gateway: 192 . 168 . 1 . 254

DNS Server: 8 . 8 . 8 . 8

3.4 Serial Port Settings

The "Serial Port" web management page allows you to set the COM port parameters, local port number, the remote server's IP address and port number for which this device is connected to, the operating mode (e.g. TCP client or TCP server), and other features including reset, link, index, and RFC2217.

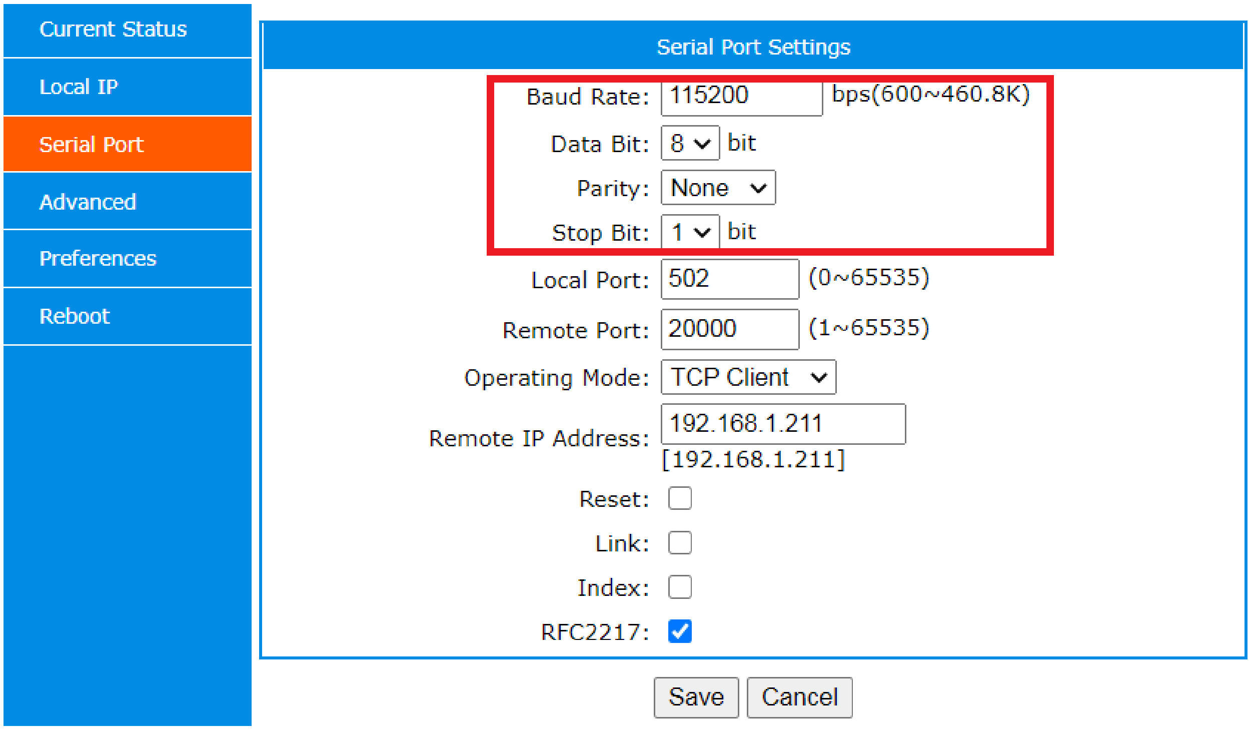
Table 2: Default COM Port and IP Port settings

Baud Rate:	115.2Kbps
Data Bit:	8
Parity:	None
Stop Bit:	1
Local IP:	192.168.1.201 (refer to Local IP settings)
Local Port:	502
Remote IP:	192.168.1.211
Remote Port:	20,000
Operating Mode:	TCP Client

3.4.1 COM Port Settings

The COM port settings must match with the RS-485 device to which the DVS-485-2 is connected, COM port settings include the following parameters:

- Baud Rate: 600bps to 460.8Kbps
- Data Bit: 7 or 8
- Parity: None, Odd, Even, Mark, Space
- Stop Bit: 1 or 2



Serial Port Settings

Baud Rate: 115200 bps(600~460.8K)

Data Bit: 8 bit

Parity: None

Stop Bit: 1 bit

Local Port: 502 (0~65535)

Remote Port: 20000 (1~65535)

Operating Mode: TCP Client

Remote IP Address: 192.168.1.211 [192.168.1.211]

Reset: □

Link: □

Index: □

RFC2217: ☑

3.4.2 Operating Mode

CommFront’s device servers offer three types of network connections: TCP socket, UDP socket, and HTTP web socket.

TCP is a connection-oriented protocol that provides reliable, ordered and error-checked delivery for network packets. For such reasons, TCP is recommended for device servers because mission-critical messages are transmitted over a crowded Ethernet network. Under TCP operating mode, device servers must first establish a client-server network connection prior to transmitting serial data over a TCP/IP network. Up to 8 clients can co-exist and communicate with the server concurrently; however, only one server can exist in a client-server network connection.

UDP is a datagram-oriented (connectionless) protocol that is used for broadcast and multicast network transmission. UDP works similar to TCP, and it also checks for data integrity at the arrival time. However, UDP protocol has no fixed order, and it discards erroneous packets.

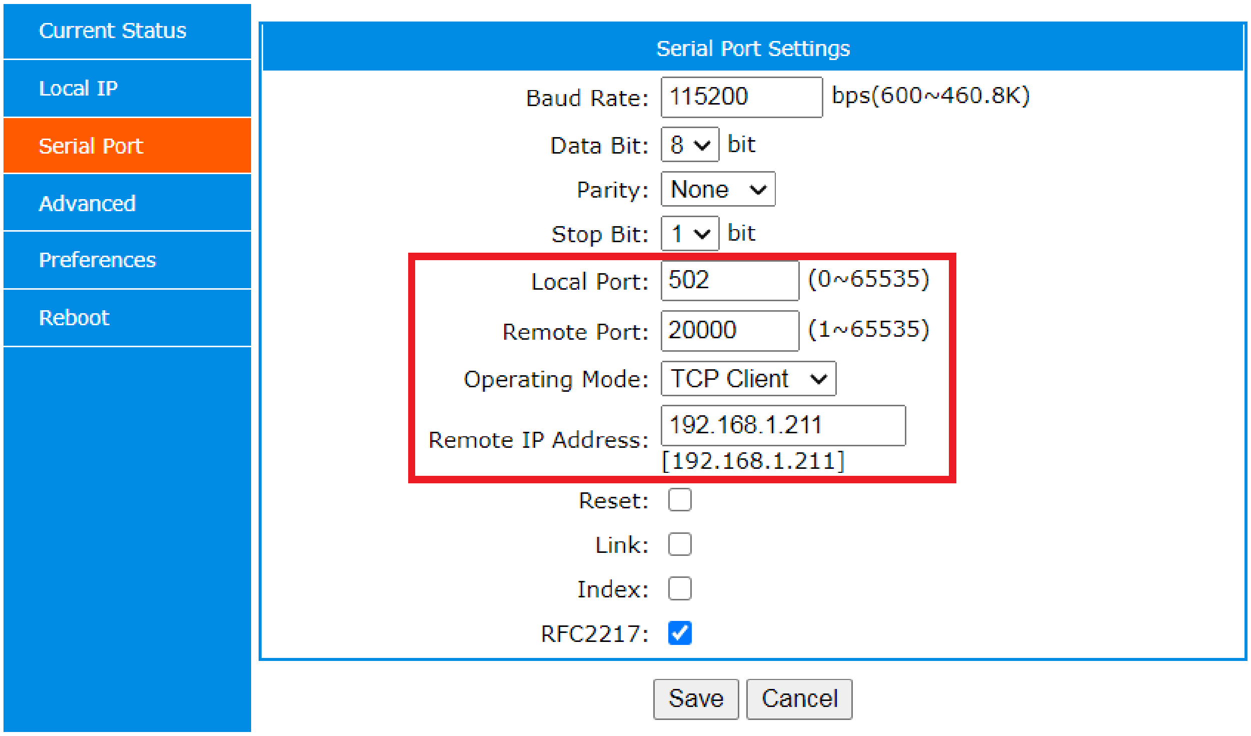
HTTPD (or HTTP) is an application-layer protocol for transmitting web contents or data between web server and the clients. HTTPD follows a classical client-server model, which means requests are initiated by the client, usually a web browser.

There are a total of five operating modes in CommFront’s device servers.

1) **TCP Client:**
Under TCP client mode, the DVS-485-2 automatically initiates a TCP network connection request to the remote server. If the connection is successful, client's IP address will be displayed under server's "current status", and if the connection is unsuccessful, the DVS-485-2 will keep on sending connection requests until it is successful. Once connected, TCP client can start to forward / receive serial data to / from server's serial port. TCP client can choose to send heartbeat (or keep-alive) packets to remote server for maintaining the network connections (see "Heartbeat Packets" for more information).

- Remote IP: Enter remote TCP server's IP address
- Remote Port: Enter remote TCP server's port number
- Local IP: *(To be set under "Local IP")*
- Local Port: Enter local port number, if the number is 0, the device will use random number for connecting to remote server.

Note: Upon TCP connection, serial data received on client's COM port will be forwarded to remote server only. Other clients will not receive the same serial data.



Serial Port Settings

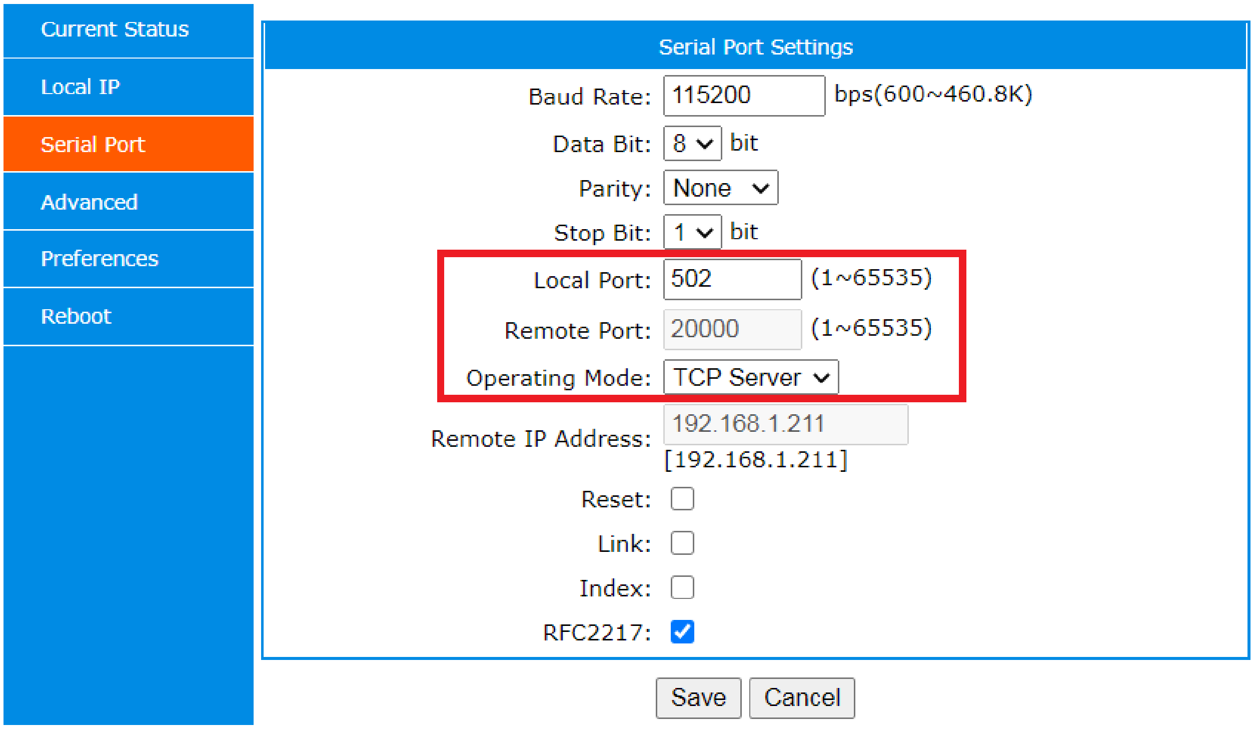
Baud Rate: 115200 bps(600~460.8K)
Data Bit: 8 bit
Parity: None
Stop Bit: 1 bit
Local Port: 502 (0~65535)
Remote Port: 20000 (1~65535)
Operating Mode: TCP Client
Remote IP Address: 192.168.1.211 [192.168.1.211]
Reset: □
Link: □
Index: □
RFC2217: ☑

2) TCP Server:

Under TCP server mode, the DVS-485-2 will monitor its local port and establish connections whenever there is a new TCP connection request. TCP server allows a maximum of 8 client connections concurrently. If the number of client connections exceed 8, you can set to disable or keep the old TCP connections (see "Disable Old Connections" for more information).

- Local IP: *(To be set under "Local IP")*
- Local Port: Enter local port number for the server. 0 is not allowed.

Note: Upon TCP connection, serial data received on the server's COM port will be forwarded to ALL clients.



Serial Port Settings

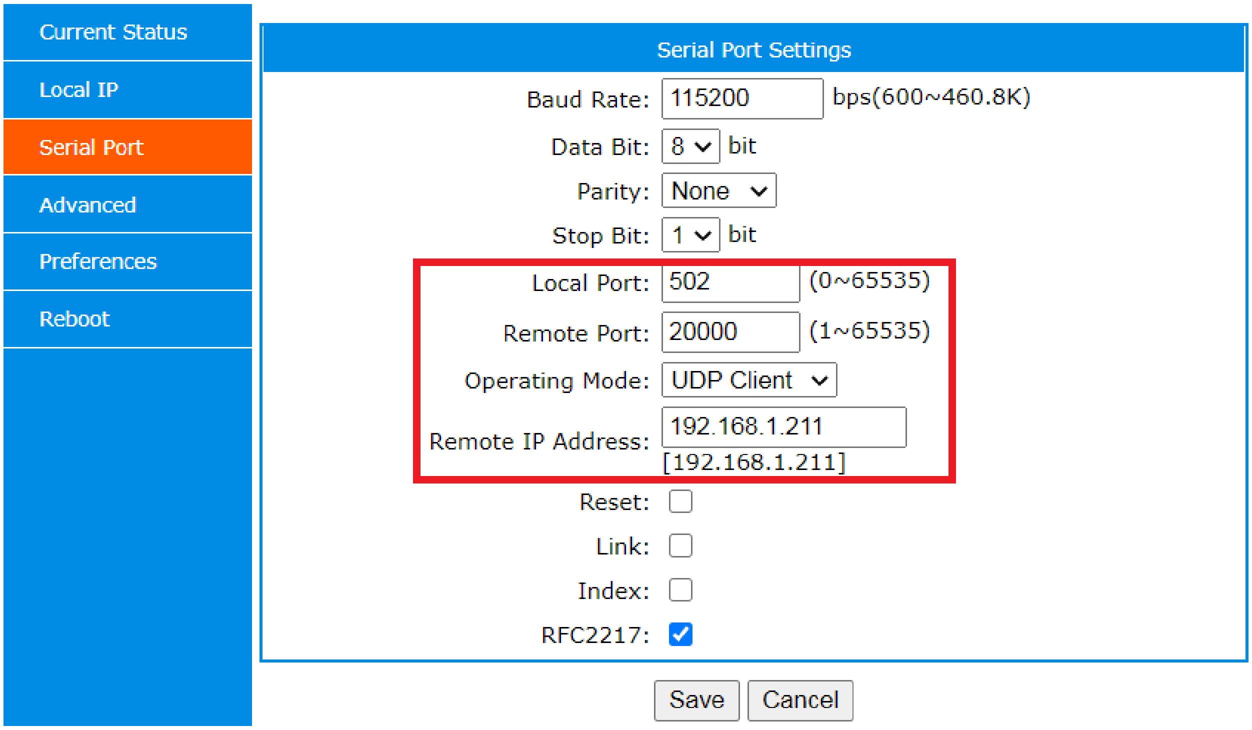
Baud Rate:	115200	bps(600~460.8K)
Data Bit:	8	bit
Parity:	None	
Stop Bit:	1	bit
Local Port:	502	(1~65535)
Remote Port:	20000	(1~65535)
Operating Mode:	TCP Server	
Remote IP Address:	192.168.1.211 [192.168.1.211]	
Reset:	□	
Link:	□	
Index:	□	
RFC2217:	☑	

3) UDP Client:

UDP is a connectionless protocol and does not require a network connection like TCP prior to transmitting serial data over Ethernet. A UDP client can start to forward serial data to the UDP server's serial port as soon as the server's IP address and port number are defined and matched.

- Remote IP: Enter remote UDP server's IP address
- Remote Port: Enter remote UDP server's port number
- Local IP: *(To be set under "Local IP")*
- Local Port: Enter local port number. If the number is 0, the device will use a random number.

Note: Serial data received on the UDP client's COM port will be forwarded to remote UDP server only. Other clients will not receive the same serial data.



Serial Port Settings

Baud Rate: 115200 bps(600~460.8K)

Data Bit: 8 bit

Parity: None

Stop Bit: 1 bit

Local Port: 502 (0~65535)

Remote Port: 20000 (1~65535)

Operating Mode: UDP Client

Remote IP Address: 192.168.1.211 [192.168.1.211]

Reset: □

Link: □

Index: □

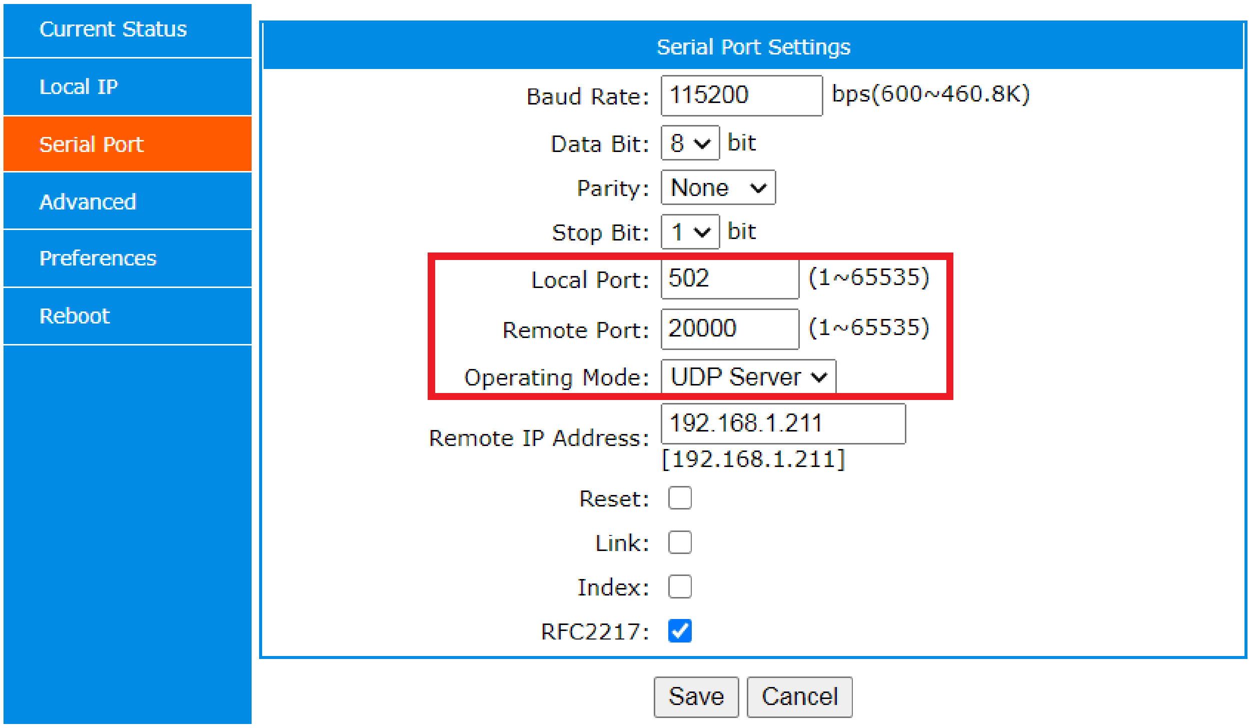
RFC2217: ☑

4) **UDP Server:**

Unlike a TCP server, which forwards its serial data to all connected clients, a UDP server forwards serial data only to the source (IP + port) of last-received data. For example, if the UDP server receives data from Client A first and then from Client B, its serial data will be forwarded to Client B only. (Client A will not receive any data from the UDP server).

- Local IP: *(To be set under "Local IP")*
- Local Port: Enter local port number for the server. 0 is not allowed.

Note: UDP server does not have any destination IP and port to forward its serial data and therefore, communications must be initiated from UDP client.



Serial Port Settings

Baud Rate: 115200 bps(600~460.8K)

Data Bit: 8 bit

Parity: None

Stop Bit: 1 bit

Local Port: 502 (1~65535)

Remote Port: 20000 (1~65535)

Operating Mode: UDP Server

Remote IP Address: 192.168.1.211 [192.168.1.211]

Reset: □

Link: □

Index: □

RFC2217: ☑

5) HTTPD Client:

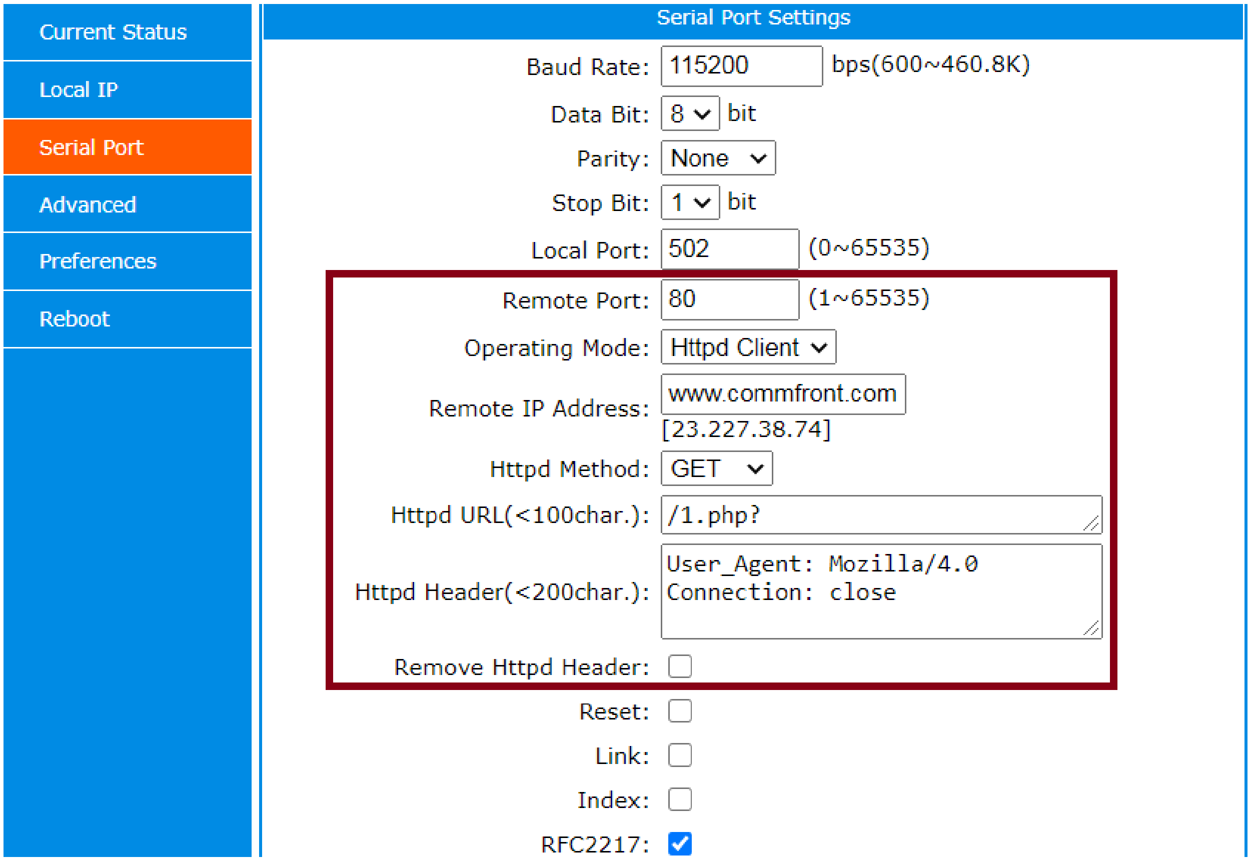
The DVS-485-2 uses HTTPD client (or HTTP client), typically used to communicate with Apache HTTP server, to create a virtual connection between its serial port and a HTTP server, which can be used for posting or fetching serial data to / from HTTP server. There are two types of HTTP messages: requests (client to server) and responses (server to client), each with its own format.

Requests consist of an HTTP method (GET or POST), a path of the HTTP server's resource destination, the version of the HTTP protocol, the optional headers that convey additional information, and a body for the data.

Responses consist of the version of the HTTP protocol they follow, a status code (indicating if the request was successful or not and why), a status message, HTTP headers, and a body containing the fetched data.

-

- Remote IP: Enter remote IP address or domain name.
- Remote Port: Enter remote HTTP server's port number, usually 80.
- Httpd Method: GET or POST. Use GET if client wants to fetch data from HTTP server. Use POST if client wants to post data to HTTP server.
- Httpd URL: Enter the path of the HTTP server's resource destination.
- Httpd Header: Enter additional information for the HTTP server.
- Remove Httpd Header: Enable or disable removal of Httpd header.



Serial Port Settings

Baud Rate: 115200 bps(600~460.8K)
Data Bit: 8 bit
Parity: None
Stop Bit: 1 bit
Local Port: 502 (0~65535)
Remote Port: 80 (1~65535)
Operating Mode: Httpd Client
Remote IP Address: www.commfront.com [23.227.38.74]
Httpd Method: GET
Httpd URL(<100char.): /1.php?
Httpd Header(<200char.): User_Agent: Mozilla/4.0
Connection: close
Remove Httpd Header: □
Reset: □
Link: □
Index: □
RFC2217: ☑

3.4.3 Reset

When the "Reset" function is enabled, the DVS-485-2 will automatically restart after 30 unsuccessful connection attempts. This performs the same function as AT command "AT+CLIENTRST".

3.4.4 Link

When the "Link" function is enabled, the "Link" pin will output a low-level voltage whenever there is a TCP connection. Otherwise, it will output a high-level voltage. This performs the same function as AT command "AT+SCSLINK".

Note: "Link" pin is on the circuit board and can only be used with PCBA.

3.4.5 Index

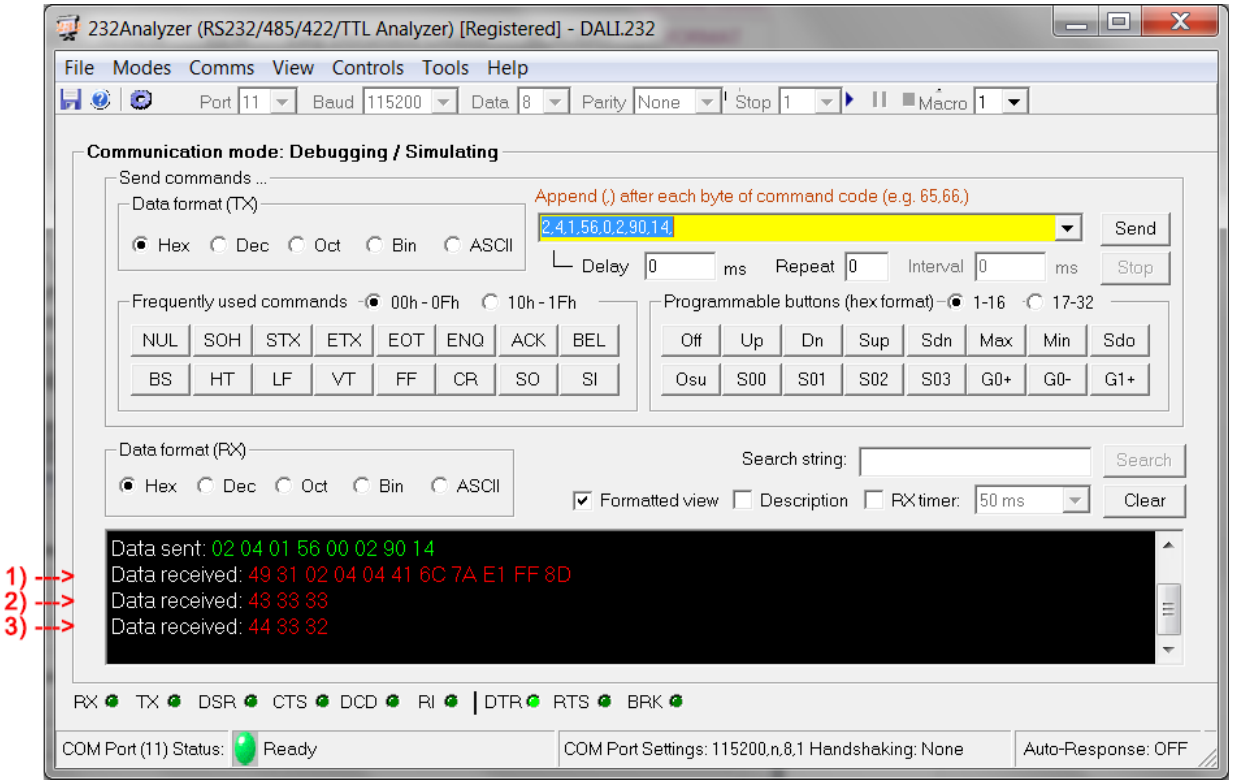
Under TCP server mode, a maximum of 8 TCP clients are allowed to be connected to the server. However, it is difficult to know the data source when there is data coming from the clients, and it is also difficult to know the current client connection status in real time. "Index" is a useful feature for TCP server. It does not just help to identify the data source but also reports the client connection status to the connected serial devices.

When the "Index" function is turned on:

1) Two additional bytes (ASCII code "I" "m": "I" stands for Incoming and "m" is the index number) are added to the beginning of the incoming packet from TCP clients. The TCP server will now be able to identify where the packet comes from.
2) When the TCP client is disconnected, it sends three bytes (ASCII code "D" "m" "n": "D" stands for Disconnect, "m" means the disconnected client's index number, and "n" means the remaining number of client connections) to the server's COM port.
3) When a TCP client is connected, it sends three bytes (ASCII code "C" "m" "n": "C" stands for Connect, "m" means the connected index number, and "n" means total number of connections) to the server's COM port.

Example:

1) Client index 1 sends in data, and it adds ASCII code "I1" (Hex code "49 31") to the beginning of the packet.
2) Client index 3 connected, and COM port received "C32" (Hex code "43 33 33").
3) Client index 3 disconnected, and COM port received "D32" (Hex code "44 33 32").



232Analyzer (RS232/485/422/TTL Analyzer) [Registered] - DALI.232

Port 11 Baud 115200 Data 8 Parity None Stop 1 Macro 1

Communication mode: Debugging / Simulating

Data format (TX): ● Hex

Append (,) after each byte of command code (e.g. 65,66,)

2,4,1,56,0,2,90,14,

Delay 0 ms Repeat 0 Interval 0 ms

Data format (RX): ● Hex ○ Dec ○ Oct ○ Bin ○ ASCII

☑ Formatted view □ Description □ RX timer: 50 ms

```
Data sent: 02 04 01 56 00 02 90 14
Data received: 49 31 02 04 04 41 6C 7A E1 FF 8D
Data received: 43 33 33
Data received: 44 33 32
```

COM Port (11) Status: Ready | COM Port Settings: 115200,n,8,1 Handshaking: None | Auto-Response: OFF

3.4.6 RFC2217

The RFC2217 feature allows the COM port settings to be dynamically changed from TCP/IP network. Such changes will take effect immediately but will be lost when the device is restarted or loses power. The operating modes that support RFC2271 are: TCP client, TCP server, UDP client, and UDP server. The RFC2217 packet does not forward to a serial port, and it consists of 8 bytes.

Table 3: RFC2217 Protocol

COM port settings	Byte 1 to 3	Byte 4 to 6 (*)	Byte 7 (***)	Byte 8 (**)
Baud, Data Bit, Parity, Stop Bit	Header	Baud Rate (Decimal to Hex)	Data Bit, Parity, Stop Bit	Checksum (Sum of Byte 4 to 7)
9600, 8, N, 1	55 AA 55	00 25 80	03	A8
19200, 8, N, 1	55 AA 55	00 4B 00	03	4E
115200, 8, N, 1	55 AA 55	01 C2 00	03	C6

Notes:

* Byte 4 to 6 is to convert the decimal number directly to a hexadecimal number, e.g. 9600 = 002580; 119200 = 004B00; 115200 = 01C200

** Byte 8 is to add together byte 4, 5, 6, and byte 7, e.g. 00+25+80+03=A8; 00+4B+00+03=4E; 01+C2+00+03 = C6

Table 4: *** Byte 7

Bit 7 & 6	Bit 5 & 4	Bit 3	Bit 2	Bit 1 & 0
Undefined	Parity	Parity (ON/OFF)	Stop Bit	Data Bit
00	00: NONE/ODD 01: EVEN 10: Mark 11: Space	00: OFF 01: ON	00: 1 Stop Bit 02: 2 Stop Bit	00: 5 Data Bit 01: 6 Data Bit 10: 7 Data Bit 11: 8 Data Bit

Example: Byte 7 for the settings of 8 Data Bits, None (Parity), and 1 Stop Bit = 03

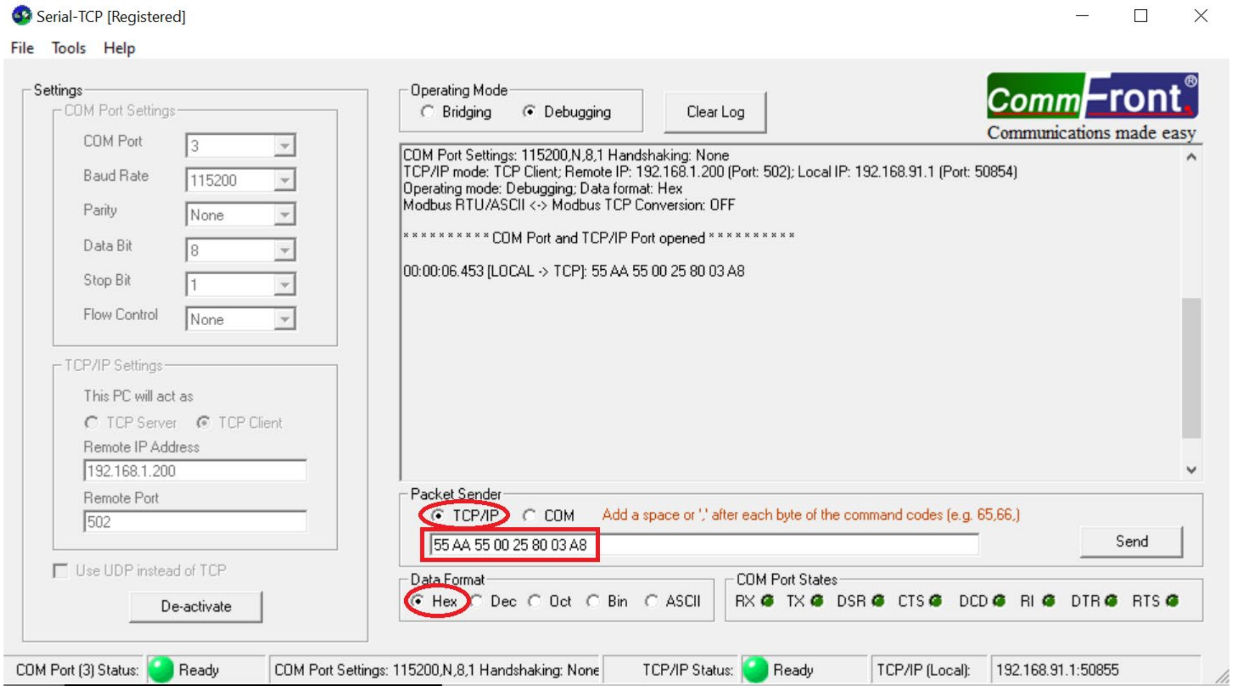
RFC2217 Command Examples:

1) The TCP/IP command for changing COM port settings to 9600, 8, N, 1 is: 55 AA 55 00 25 80 03 A8
2) The TCP/IP command for changing COM port settings to 19200, 8, N, 1 is: 55 AA 55 00 4B 00 03 4E
3) The TCP/IP command for changing COM port settings to 115200, 8, N, 1 is: 55 AA 55 01 C2 00 03 C6

Sending RFC2217 commands from CommFront's Serial-TCP software:

1) Set the device as TCP server.
2) Launch CommFront's Serial-TCP free software and type in the Device Server's IP address and Port number. Then choose "TCP Client" mode and click the "Activate" button.
3) After the connection is established, choose "TCP/IP" under Packet Sender and choose "Hex" for Data Format. Then key in and send the RFC2217's command (e.g. 55 AA 55 00 25 80 03 A8).
4) The COM port settings of the device server will then be changed to: 9600, 8, N, 1.

Note: The RFC2217 command is a device server's command, so it will not be forwarded to serial port.



Serial-TCP [Registered]

Settings — COM Port Settings: COM Port 3; Baud Rate 115200; Parity None; Data Bit 8; Stop Bit 1; Flow Control None

TCP/IP Settings: This PC will act as ○ TCP Server ● TCP Client; Remote IP Address 192.168.1.200; Remote Port 502

□ Use UDP instead of TCP

Operating Mode: ○ Bridging ● Debugging

```
COM Port Settings: 115200,N,8,1 Handshaking: None
TCP/IP mode: TCP Client; Remote IP: 192.168.1.200 (Port: 502); Local IP: 192.168.91.1 (Port: 50854)
Operating mode: Debugging; Data format: Hex
Modbus RTU/ASCII <-> Modbus TCP Conversion: OFF

********** COM Port and TCP/IP Port opened **********

00:00:06.453 [LOCAL -> TCP]: 55 AA 55 00 25 80 03 A8
```

Packet Sender: ● TCP/IP ○ COM — Add a space or ',' after each byte of the command codes (e.g. 65,66,)

55 AA 55 00 25 80 03 A8

Data Format: ● Hex ○ Dec ○ Oct ○ Bin ○ ASCII

COM Port (3) Status: Ready | COM Port Settings: 115200,N,8,1 Handshaking: None | TCP/IP Status: Ready | TCP/IP (Local): 192.168.91.1:50855

3.5 Advanced Settings

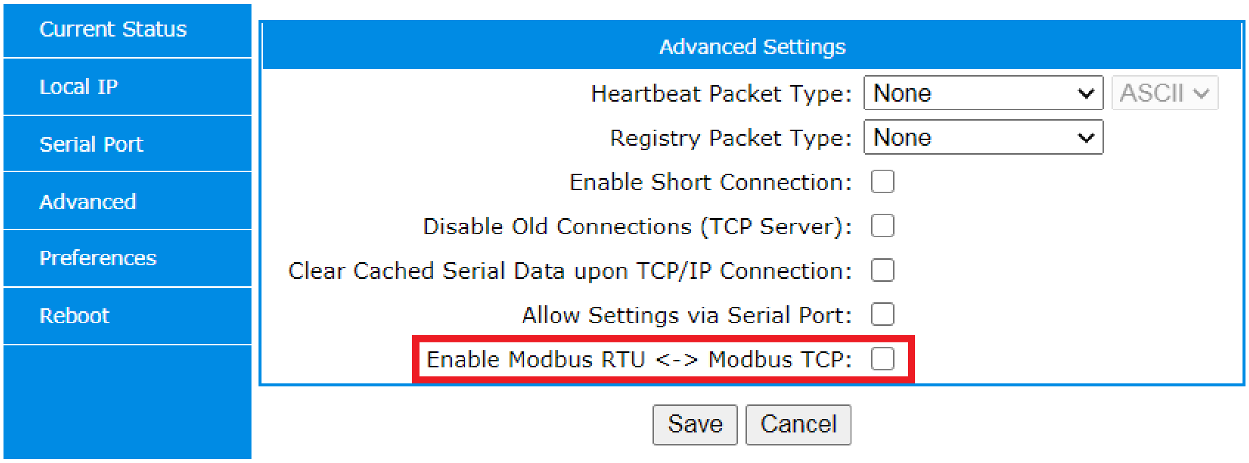
The "Advanced" web management page allows you to set advanced features such as Modbus RTU to Modbus TCP conversion, Heartbeat Packet, Registry Packet, and short connection.

Table 5: Default Advanced Settings

Heartbeat Packet Type:	None
Registry Packet Type:	None
Enable Short Connection:	Unchecked
Disable Old Connections:	Unchecked
Clear Cached Serial Data upon TCP/IP Connection:	Unchecked
Allow Settings via Serial Port:	Unchecked
Enable Modbus RTU <-> Modbus TCP	Unchecked

3.5.1 Modbus RTU to Modbus TCP

When selected, it will turn the DVS-485-2 into a Modbus RTU to Modbus TCP gateway or a Modbus RTU to Modbus TCP protocol converter. Modbus RTU data received from the serial port will be converted to Modbus TCP and transmitted over TCP/IP network. Likewise, Modbus TCP data received from the network port will be converted to Modbus RTU and transmitted to serial port. This feature works for TCP Server and TCP Client modes.



Advanced Settings

Heartbeat Packet Type: None ASCII

Registry Packet Type: None

Enable Short Connection: □

Disable Old Connections (TCP Server): □

Clear Cached Serial Data upon TCP/IP Connection: □

Allow Settings via Serial Port: □

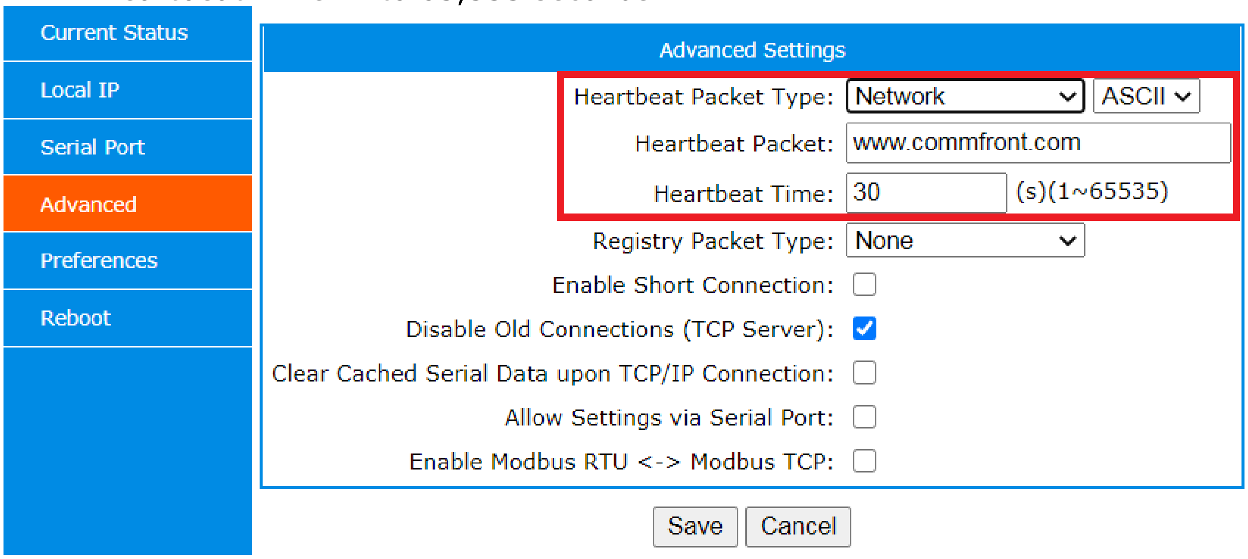
Enable Modbus RTU <-> Modbus TCP: □

3.5.2 Heartbeat Packet

The heartbeat packet (or "keep-alive" packet) can be sent either to a network port or a serial port. The network heartbeat packet is used for maintaining network connections and to prevent dead links. It works for TCP client and UDP client modes. The serial heartbeat packet can be used for routine commands sent to the serial devices.

- Heartbeat Packet Type: None, Serial, Network
- Heartbeat Packet: ASCII string or HEX code

- Heartbeat Time: 1 to 65,535 seconds



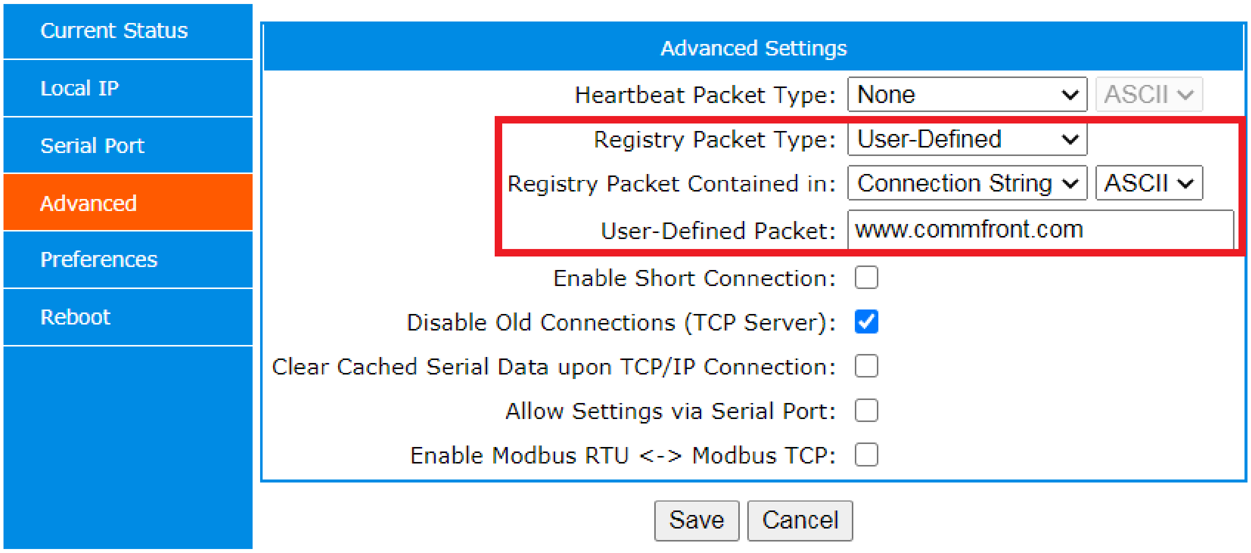
Advanced Settings

Heartbeat Packet Type: Network ASCII
Heartbeat Packet: www.commfront.com
Heartbeat Time: 30 (s)(1~65535)
Registry Packet Type: None
Enable Short Connection: □
Disable Old Connections (TCP Server): ☑
Clear Cached Serial Data upon TCP/IP Connection: □
Allow Settings via Serial Port: □
Enable Modbus RTU <-> Modbus TCP: □

3.5.3 Registry Packet

Registry packets are used for establishing connections or identifying data sources for IIoT cloud applications. Registry Packet is used for TCP client mode.

- Registry Packet Type: None, MAC address, User-Defined, CF Cloud
- Registry Packet Contained in: Connection String, Data String, Both
- User-Defined Packet: ASCII string or HEX code



Advanced Settings

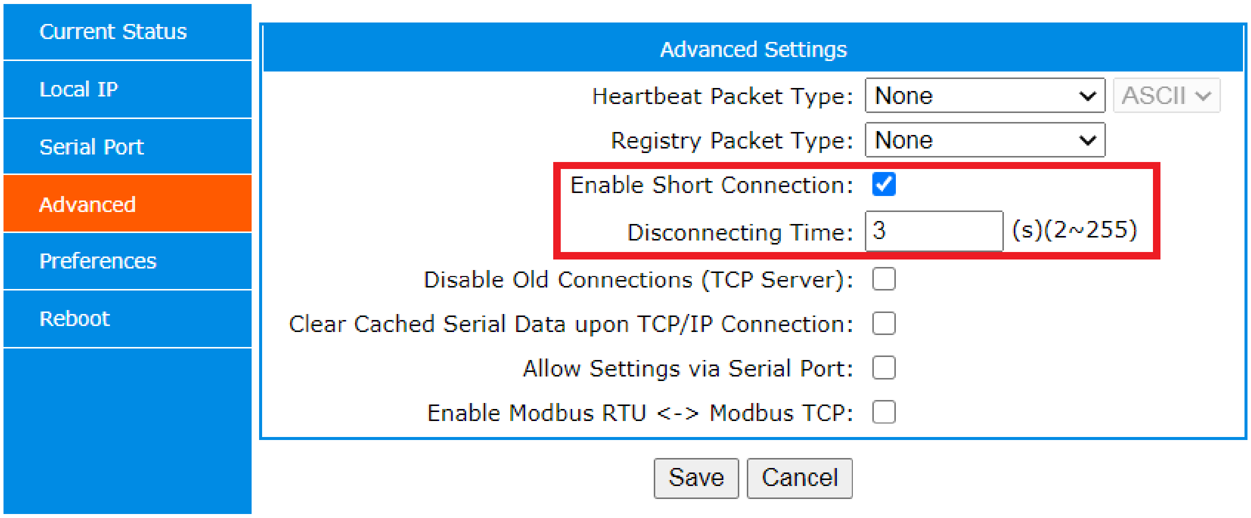
Heartbeat Packet Type: None ASCII
Registry Packet Type: User-Defined
Registry Packet Contained in: Connection String ASCII
User-Defined Packet: www.commfront.com
Enable Short Connection: □
Disable Old Connections (TCP Server): ☑
Clear Cached Serial Data upon TCP/IP Connection: □
Allow Settings via Serial Port: □
Enable Modbus RTU <-> Modbus TCP: □

3.5.4 Short Connection

“Shot connection” is used for TCP client mode and allows users to save the server’s resources by disabling TCP connections whenever data transmission (client to server) is completed. Client will automatically initiate a new TCP connection when there is data coming from the serial port, and this TCP connection will be disconnected when data transmission is completed and the disconnecting timeout is reached.

- Disconnecting Time: 2 to 255 seconds

Note: Short connection is useful if the number of client connections exceed the maximum number allowed (range from 1 to 8, to be set under “Preferences”).

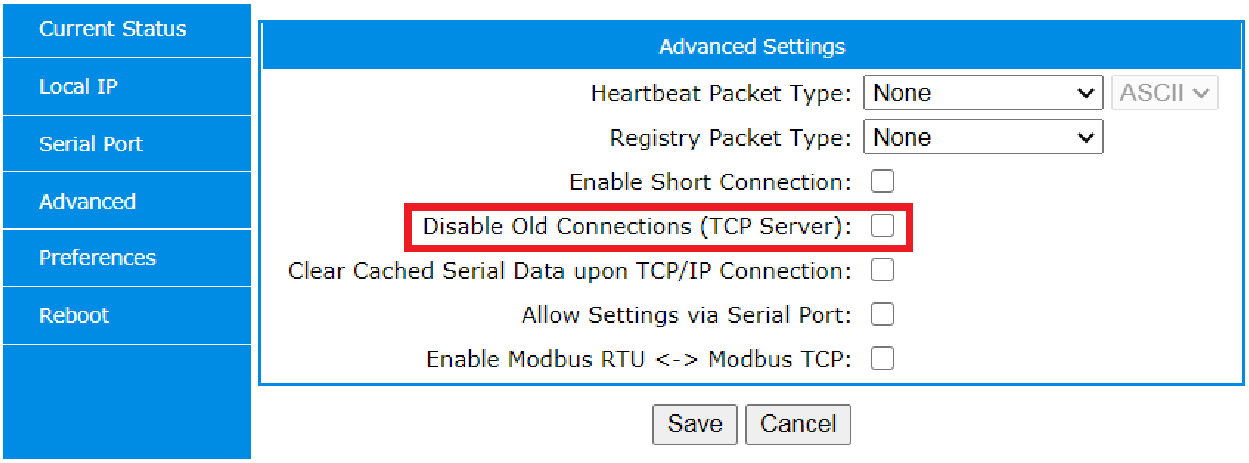


Advanced Settings

Heartbeat Packet Type: None ASCII
Registry Packet Type: None
Enable Short Connection: ☑
Disconnecting Time: 3 (s)(2~255)
Disable Old Connections (TCP Server): □
Clear Cached Serial Data upon TCP/IP Connection: □
Allow Settings via Serial Port: □
Enable Modbus RTU <-> Modbus TCP: □

3.5.5 Disable Old Connections (TCP Server)

In TCP Server mode, if the number of client connections exceeds the maximum number allowed (range from 1 to 8, to be set under “Preferences”), there are two ways to handle existing and new connections: disable old connections or deny new connections. If the “Disable Old Connections” checkbox is selected, the oldest connection will be killed if there is a new client connection request.

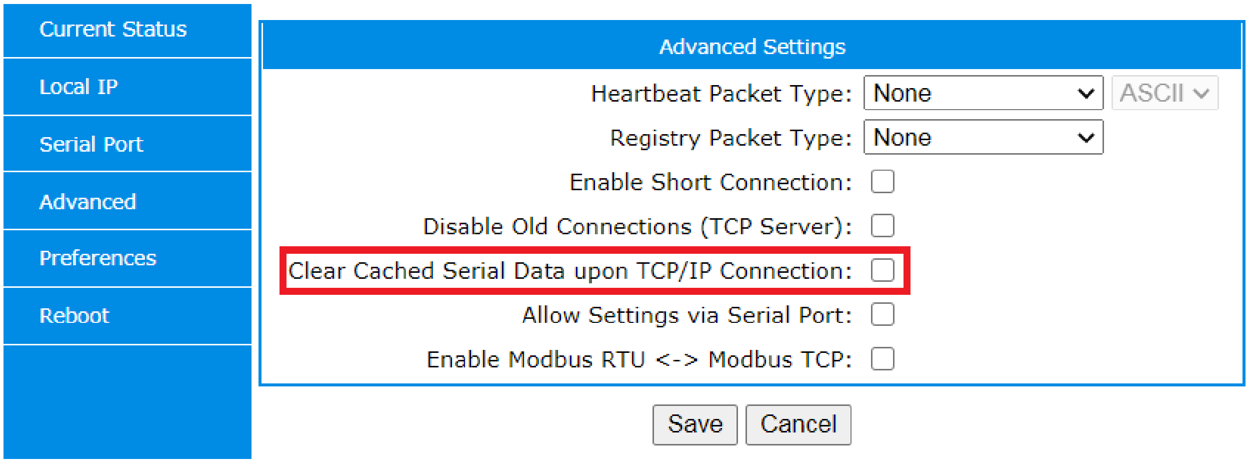


Advanced Settings

Heartbeat Packet Type: None ASCII
Registry Packet Type: None
Enable Short Connection: □
Disable Old Connections (TCP Server): □
Clear Cached Serial Data upon TCP/IP Connection: □
Allow Settings via Serial Port: □
Enable Modbus RTU <-> Modbus TCP: □

3.5.6 Clear Cached Serial Data upon TCP/IP Connection

Prior to a successful TCP connection, data received from a serial port will be cached in DVS-485-2’s serial buffer (up to 2KB). The user can choose to clear or keep this cached serial data. Note: This feature does not work for Httpd Client mode or TCP Client mode with the “short connection” enabled.



Advanced Settings

Heartbeat Packet Type: None ASCII

Registry Packet Type: None

Enable Short Connection: □

Disable Old Connections (TCP Server): □

Clear Cached Serial Data upon TCP/IP Connection: □

Allow Settings via Serial Port: □

Enable Modbus RTU <-> Modbus TCP: □

3.5.7 Allow Settings via Serial Port

This feature allows the user to establish settings via a serial port; however, it is not in use currently.

3.6 Preferences

The “Preferences” web management page allows users to set or change the device name, web server port, user name, password, maximum number of client connections, and auto-restart timeout.

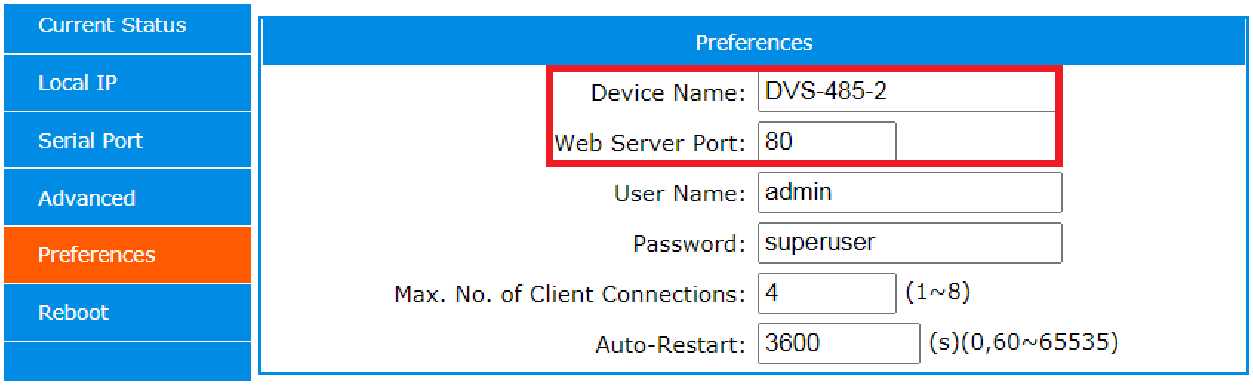
Table 6: Default Preferences Settings

Device Name:	DVS-485-2
Web Server Port:	80
User Name:	admin
Password:	superuser
Max. No. of Client Connections:	4
Auto-Restart Timeout:	3600s

3.6.1 Device Name and Web Server Port

Users can set or change device name and web server port for device management and/or security purposes.

- Device Name: Set or change device name (max. 14 characters)
- Web Server Port: Set or change web server port (max. 5-digit number)



Preferences

Device Name: DVS-485-2

Web Server Port: 80

User Name: admin

Password: superuser

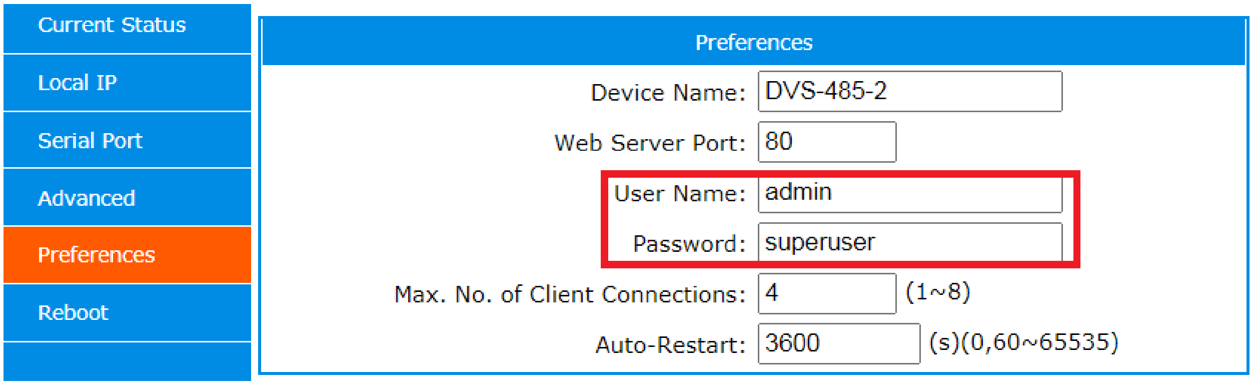
Max. No. of Client Connections: 4 (1~8)

Auto-Restart: 3600 (s)(0,60~65535)

3.6.2 User Name and Password

Users can set or change user name and password for device management and/or security purposes.

- User Name: Set or change user name (max. 5 characters)
- Password: Set or change password (max. 5 characters)



Preferences

Device Name: DVS-485-2

Web Server Port: 80

User Name: admin

Password: superuser

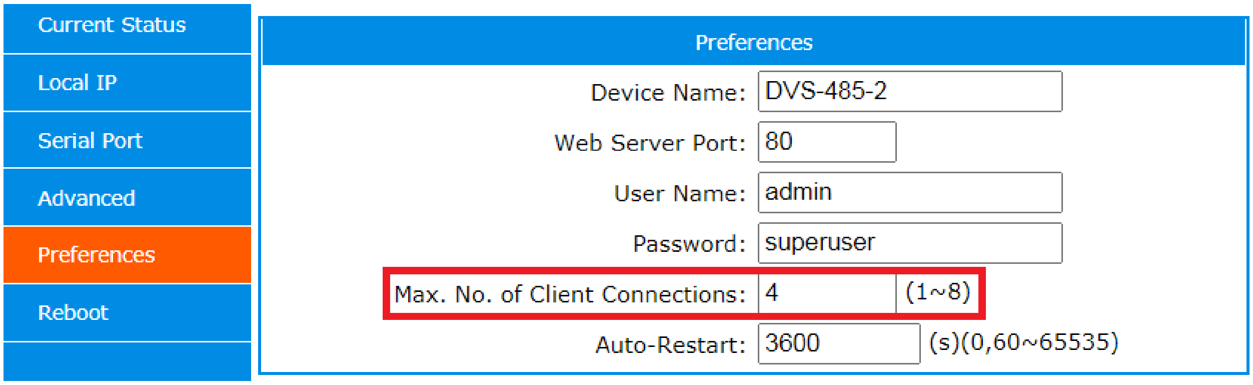
Max. No. of Client Connections: 4 (1~8)

Auto-Restart: 3600 (s)(0,60~65535)

3.6.3 Maximum Number of Client Connections

This feature works for TCP Server mode, and the number ranges from 1 to 8. When the number of client connections exceeds 4 (default), it is recommended to limit the overall data throughput to 2.5KB/s for smooth operations (e.g. reduce the serial port’s baud rate to 9,600bps for TCP clients or apply the “short connection” feature).

- Maximum Number of Client Connections: 1 to 8



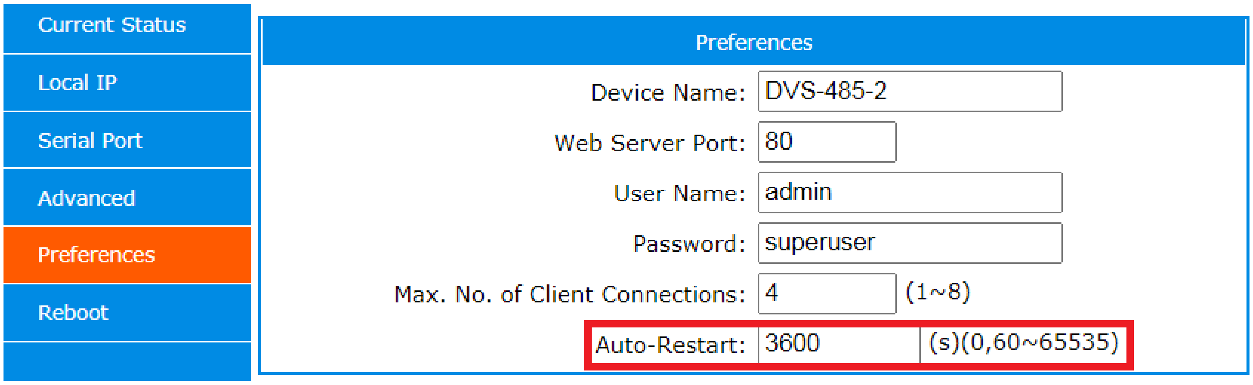
Preferences

Device Name:	DVS-485-2	
Web Server Port:	80	
User Name:	admin	
Password:	superuser	
Max. No. of Client Connections:	4	(1~8)
Auto-Restart:	3600	(s)(0,60~65535)

3.6.4 Auto-Restart Timeout

This feature allows the user to set the "Auto-Restart" timeout. The device will automatically restart when timeout is reached, which helps to prevent dead links due to long idle time.

- Auto-Restart Timeout: 0 or 60 to 65,535 seconds (0 = no auto-restart)



Preferences

Device Name:	DVS-485-2	
Web Server Port:	80	
User Name:	admin	
Password:	superuser	
Max. No. of Client Connections:	4	(1~8)
Auto-Restart:	3600	(s)(0,60~65535)

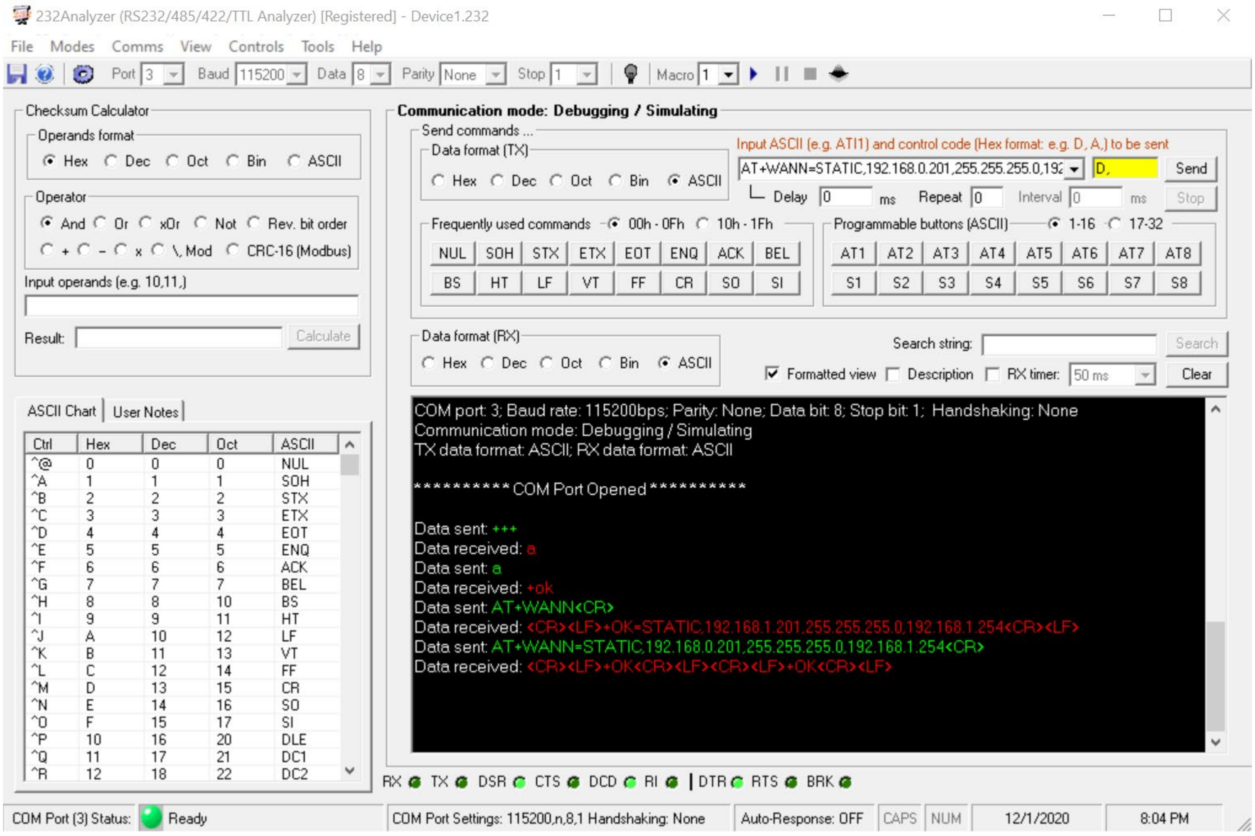
4.0 TROUBLESHOOTING

4.1 Software Tools

CommFront offers two software tools for testing and troubleshooting the device server: the awarding-winning 232Analyzer protocol analyzer software and the advanced Serial to TCP/IP bridging software. From the simplest ping and loopback tests, to a more complicated Modbus RTU and Modbus TCP tests, and to the most challenging IIoT edge applications, users can not only test a device server and make it work with a minimum effort but also combine and use our industry-proven hardware and software products for real-world projects.
Download link: https://www.commfront.com/pages/downloads

4.1.1 The 232Analyzer Protocol Analyzer Software

The 232Analyzer is an award-winning, easy-to-use, advanced RS232/RS485/RS422/ TTL/CMOS analyzer software that allows users to control, monitor, and analyze serial port activities in any data format (ASCII, hexadecimal, decimal, octal, or binary). It comes with powerful features including Programmable Buttons, Automated Responses, and Macros. The 232Analyzer supports all major Windows Operating Systems including Windows 10, 8, 7, Vista, and XP, both 32 and 64 bits.



232Analyzer (RS232/485/422/TTL Analyzer) [Registered] - Device1.232

Port 3 Baud 115200 Data 8 Parity None Stop 1 Macro 1

Checksum Calculator
Operands format: ● Hex ○ Dec ○ Oct ○ Bin ○ ASCII
Operator: ● And ○ Or ○ xOr ○ Not ○ Rev. bit order ○ + ○ - ○ x ○ \, Mod ○ CRC-16 (Modbus)
Input operands (e.g. 10,11,)
Result:

Communication mode: Debugging / Simulating
Send commands ...
Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
AT+WANN=STATIC,192.168.0.201,255.255.255.0,192 D,
Delay 0 ms Repeat 0 Interval 0 ms
Frequently used commands ● 00h - 0Fh ○ 10h - 1Fh
Programmable buttons (ASCII) ● 1-16 ○ 17-32
Data format (RX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Search string:
☑ Formatted view □ Description □ RX timer: 50 ms

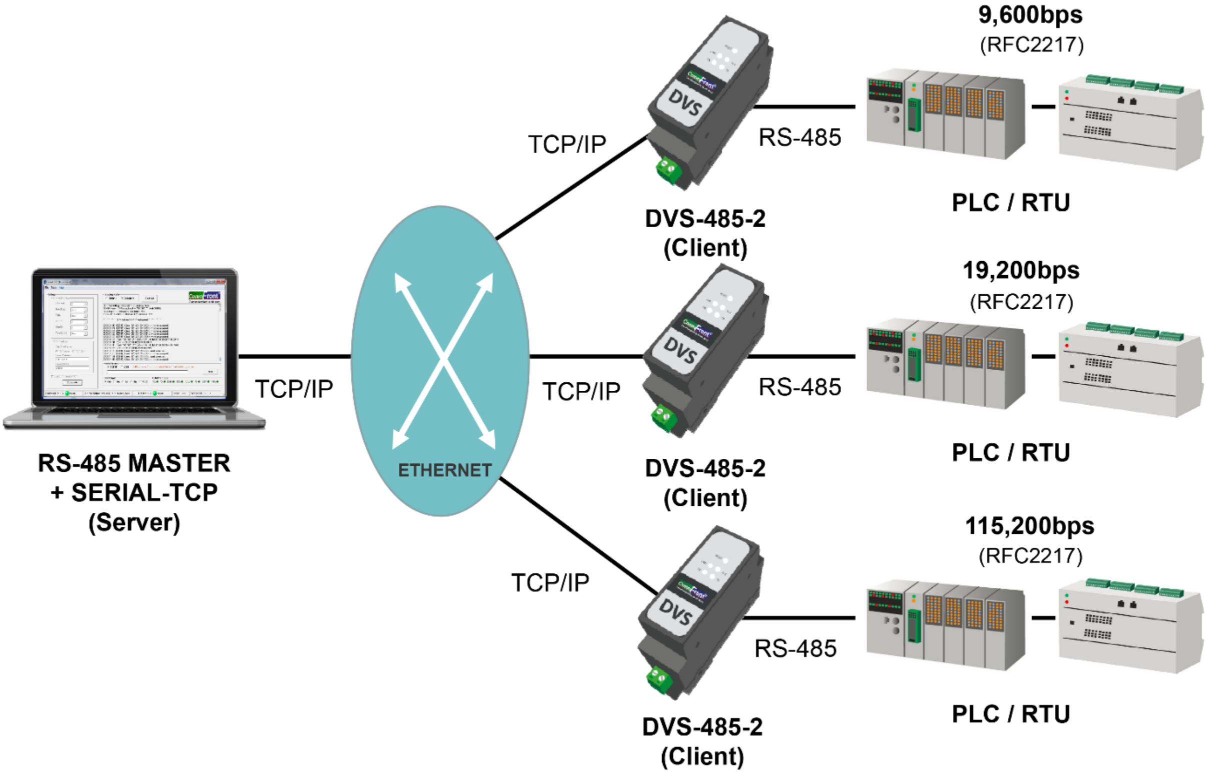
Ctrl	Hex	Dec	Oct	ASCII
^@	0	0	0	NUL
^A	1	1	1	SOH
^B	2	2	2	STX
^C	3	3	3	ETX
^D	4	4	4	EOT
^E	5	5	5	ENQ
^F	6	6	6	ACK
^G	7	7	7	BEL
^H	8	8	10	BS
^I	9	9	11	HT
^J	A	10	12	LF
^K	B	11	13	VT
^L	C	12	14	FF
^M	D	13	15	CR
^N	E	14	16	SO
^O	F	15	17	SI
^P	10	16	20	DLE
^Q	11	17	21	DC1
^R	12	18	22	DC2

```
COM port: 3; Baud rate: 115200bps; Parity: None; Data bit: 8; Stop bit: 1;  Handshaking: None
Communication mode: Debugging / Simulating
TX data format: ASCII; RX data format: ASCII

********** COM Port Opened **********

Data sent: +++
Data received: a
Data sent: a
Data received: +ok
Data sent: AT+WANN<CR>
Data received: <CR><LF>+OK=STATIC,192.168.1.201,255.255.255.0,192.168.1.254<CR><LF>
Data sent: AT+WANN=STATIC,192.168.0.201,255.255.255.0,192.168.1.254<CR>
Data received: <CR><LF>+OK<CR><LF><CR><LF>+OK<CR><LF>
```

COM Port (3) Status: Ready | COM Port Settings: 115200,n,8,1 Handshaking: None | Auto-Response: OFF | CAPS | NUM | 12/1/2020 | 8:04 PM

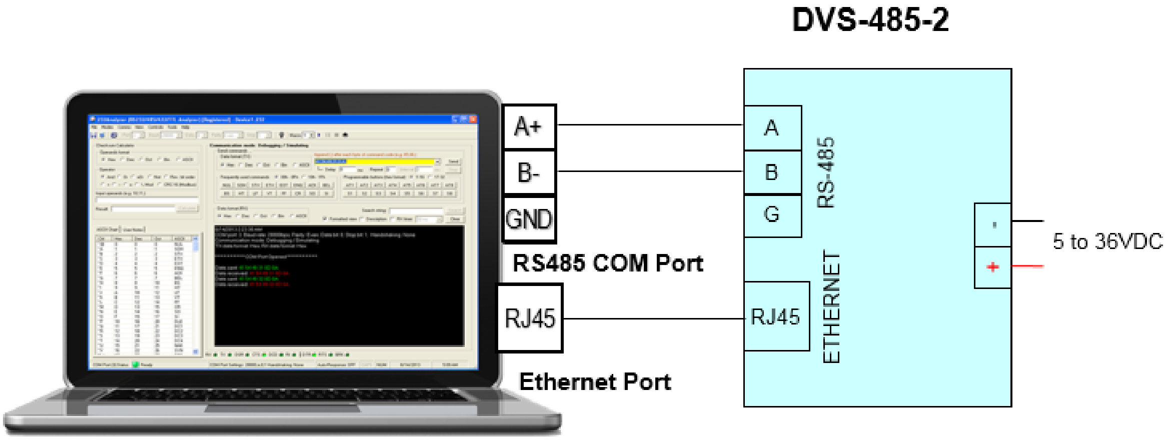
4.1.2 The Serial-TCP software

The Serial-TCP software bridges existing Serial COM port and TCP/IP without the need to change existing protocol. The applications of the Serial-TCP program include acting as a device server and/or Modbus RTU/ASCII to Modbus TCP protocol converter; extending serial communication distance via TCP/IP; replacing existing COM and TCP hardware devices; expanding the capability of existing COM and TCP/IP control software; and analyzing TCP/IP and COM network with the built-in IIoT "Packet Sender", "Ping", "Resolve IP Address/Host Name", "COM Port State LEDs", "Debugging" and "Checksum Calculator" tools. The Serial-TCP program is also very useful when converting exiting RS232/RS485/RS422 serial networks (e.g. Modbus RTU) into TCP/IP networks (e.g. Modbus TCP). The Serial-TCP supports all major Windows OS including Windows 10, 8, 7, Vista, and XP, both 32 and 64 bits.



4.2 Self Tests

4.2.1 Connecting the Power and Host PC

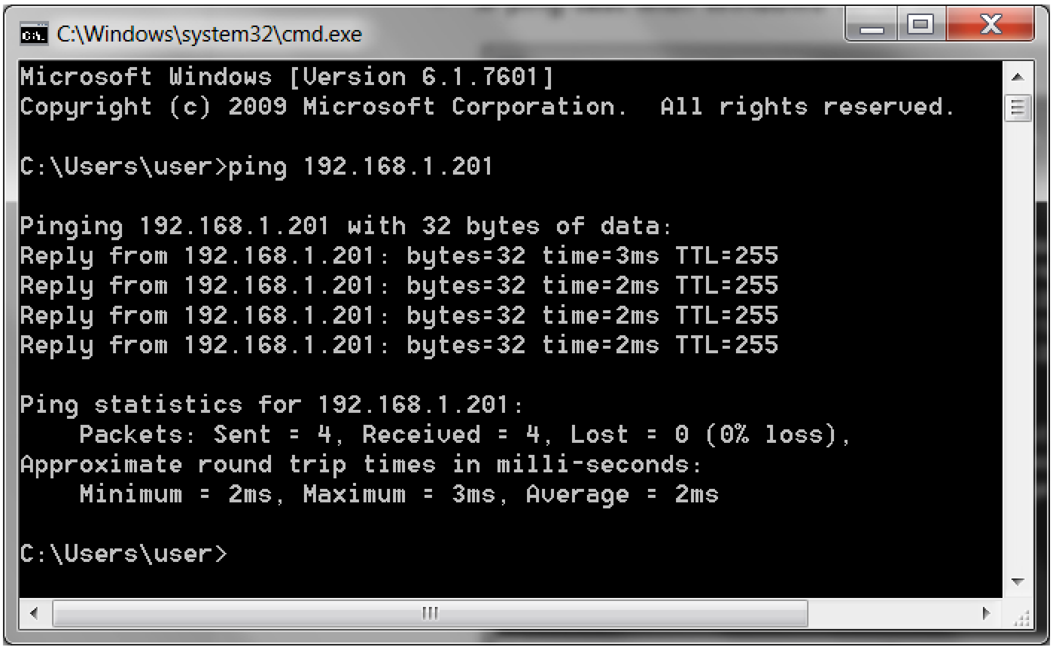


- Connect an external regulated DC power supply to the DC input (5 to 36V DC)
- Connect the DVS-485-2 to the PC's COM port and Ethernet port.

4.2.2 Ping Test

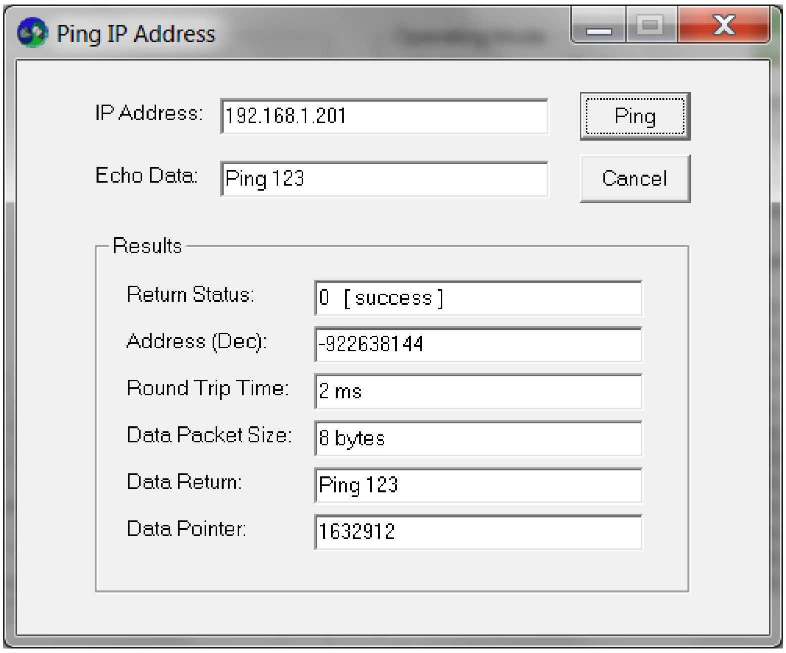
Ping is a method to test whether a particular host is reachable across an IP network. A ping measures the round-trip time for packets to be sent from the local host to a destination device (e.g. a computer or a device server) and back. Use the Windows CMD tools or CommFront Serial-TCP software to perform a ping test.

- A ping test with Windows' CMD tools



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user>ping 192.168.1.201

Pinging 192.168.1.201 with 32 bytes of data:
Reply from 192.168.1.201: bytes=32 time=3ms TTL=255
Reply from 192.168.1.201: bytes=32 time=2ms TTL=255
Reply from 192.168.1.201: bytes=32 time=2ms TTL=255
Reply from 192.168.1.201: bytes=32 time=2ms TTL=255

Ping statistics for 192.168.1.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\user>
```

- A ping test with CommFront's Serial-TCP software



IP Address: 192.168.1.201

Echo Data: Ping 123

Return Status: 0 [success]

Address (Dec): -922638144

Round Trip Time: 2 ms

Data Packet Size: 8 bytes

Data Return: Ping 123

Data Pointer: 1632912

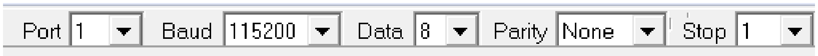
4.2.3 Setting up a device server with AT Commands

By using CommFront's 232Analyzer software, users can quickly set up the device server such as network and serial port parameters with AT commands via serial COM port.

Assume that your PC is at the same sub network of "172.16.0.x".

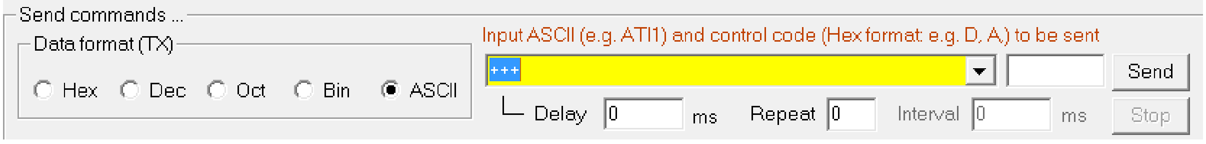
1) Download and install the free version of CommFront's 232Analyzer software (download link: https://www.commfront.com/pages/downloads).

2) Run the 232Analyzer software and set the COM port settings to 115.2Kbps, 8, N, 1. Then select "ASCII" for both TX and RX data formats.



Port 1 Baud 115200 Data 8 Parity None Stop 1

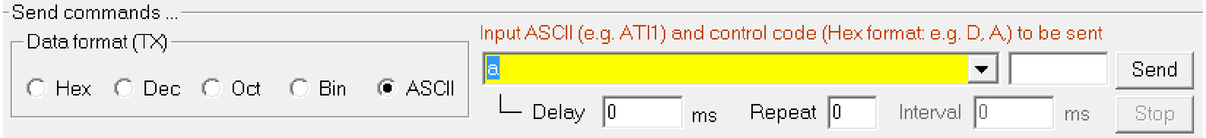
3) Send ASCII command: **+++**



Send commands ...
Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
+++
Send
Delay 0 ms Repeat 0 Interval 0 ms Stop

4) The DVS-485-2 responds with an ASCII code: **a**

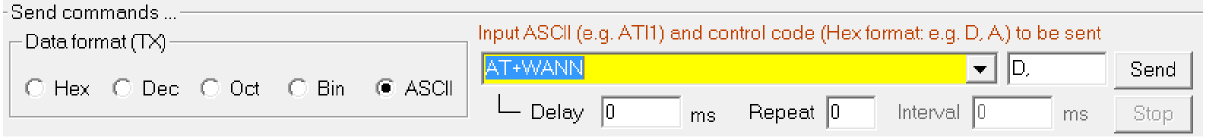
5) Send ASCII command within 3 seconds: **a**



Send commands ...
Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
a
Send
Delay 0 ms Repeat 0 Interval 0 ms Stop

6) The DVS-485-2 responds with an ASCII code, and it now enters into "AT Command" mode: **+ok**

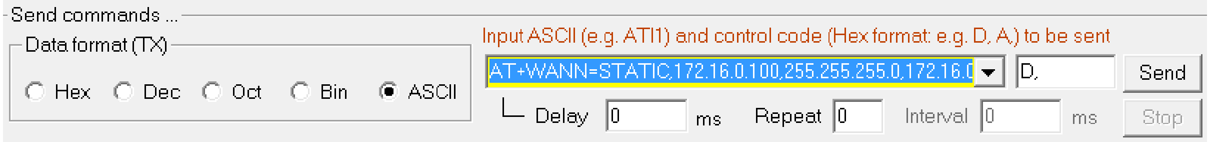
7) Check current network settings: **AT+WANN<CR>**
Note: <CR> equals to "D" in Hex code, remember to enter "D," into the second box below



Send commands ...
Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
AT+WANN
D,
Send
Delay 0 ms Repeat 0 Interval 0 ms Stop

8) The DVS-485-2 responds with the current network settings:
<CR><LF>+OK=STATIC,192.168.1.201,255.255.255.0,192.168.1.254<CR><LF>

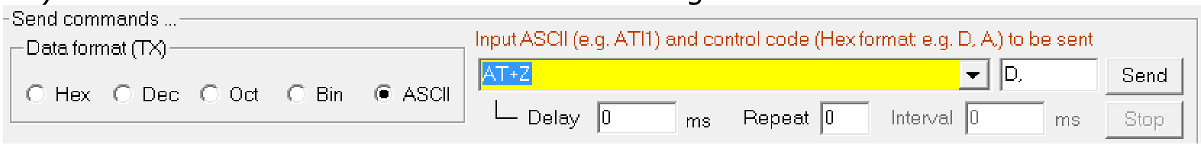
9) Change network settings to the same sub network as your PC, e.g.: **AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0.1 <CR>**



Send commands ...
Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0
D,
Send
Delay 0 ms Repeat 0 Interval 0 ms Stop

10) The DVS-485-2 responds with a confirmation code: **+ok**

11) Restart the DVS-485-2 for the new settings to take effect: **AT+Z<CR>**


Send commands ...
Data format (TX)
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent
Send
Stop

○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

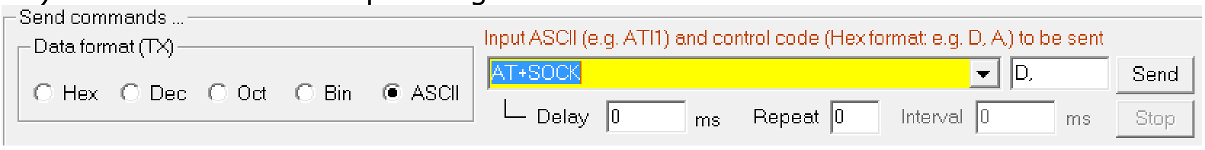
AT+Z | D,

Delay 0 ms Repeat 0 Interval 0 ms

12) The DVS-485-2 is now at the same sub network as your PC.

Note: Settings below are illustrated with the default IP settings. You may also go to web management for further configuration (see "Web Management").

13) Check the current operating mode: **AT+SOCK<CR>**


Send commands ...
Data format (TX)
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent
Send
Stop

○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

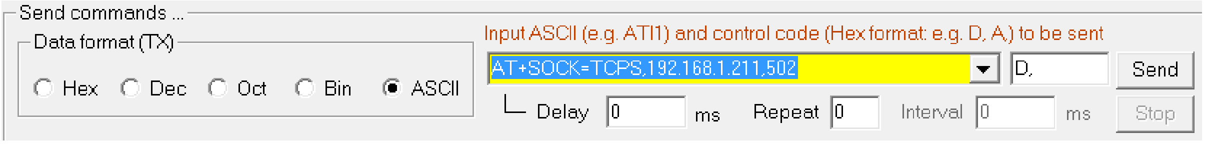
AT+SOCK | D,

Delay 0 ms Repeat 0 Interval 0 ms

14) The DVS-485-2 responds with the current operating mode:
<CR><LF>+OK=TCPC,192.168.1.211,502<CR><LF>

15) The DVS-485-2 responds with a confirmation code: **+ok**

16) Set the operating mode as "TCP Server":
AT+SOCK=TCPS,192.168.1.211,502<CR>

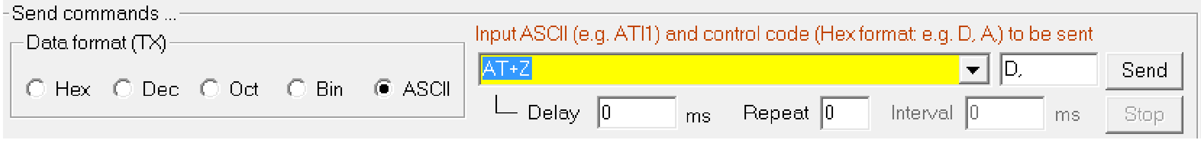

Send commands ...
Data format (TX)
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent
Send
Stop

○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

AT+SOCK=TCPS,192.168.1.211,502 | D,

Delay 0 ms Repeat 0 Interval 0 ms

17) Restart the DVS-485-2 for the new settings to take effect: **AT+Z<CR>**


Send commands ...
Data format (TX)
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A) to be sent
Send
Stop

○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

AT+Z | D,

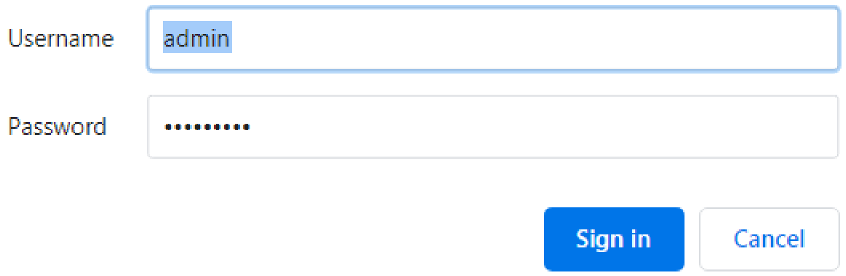
Delay 0 ms Repeat 0 Interval 0 ms

Note: For more detailed information about AT Commands, please refer to Appendix: AT Commands Set.

4.2.4 Setting Up Device Sever with Web Management Tools

Besides the AT commands, users can also log on to the device server's built-in web server and manage the settings there.

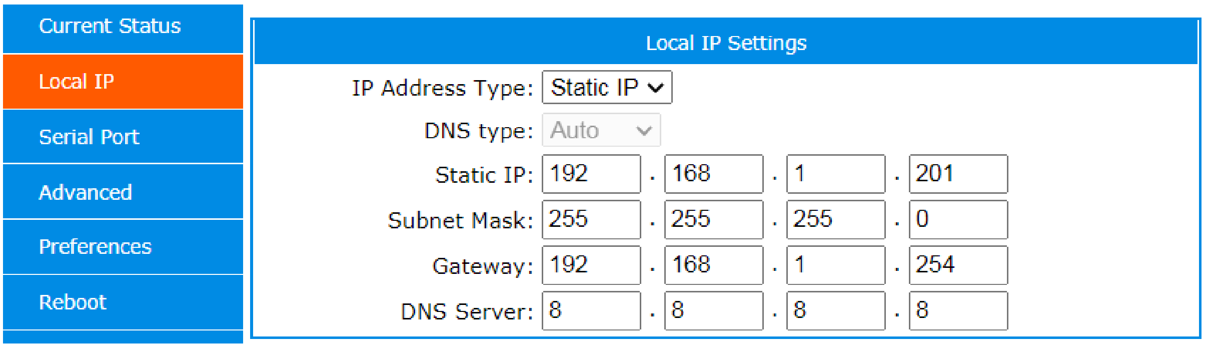
1) Log onto Web Management. A user name and password are required.
   - default user name: admin
   - default password: superuser



Username: admin

Password: •••••••••

2) Go to "Local IP" and change IP address and other parameters as necessary.



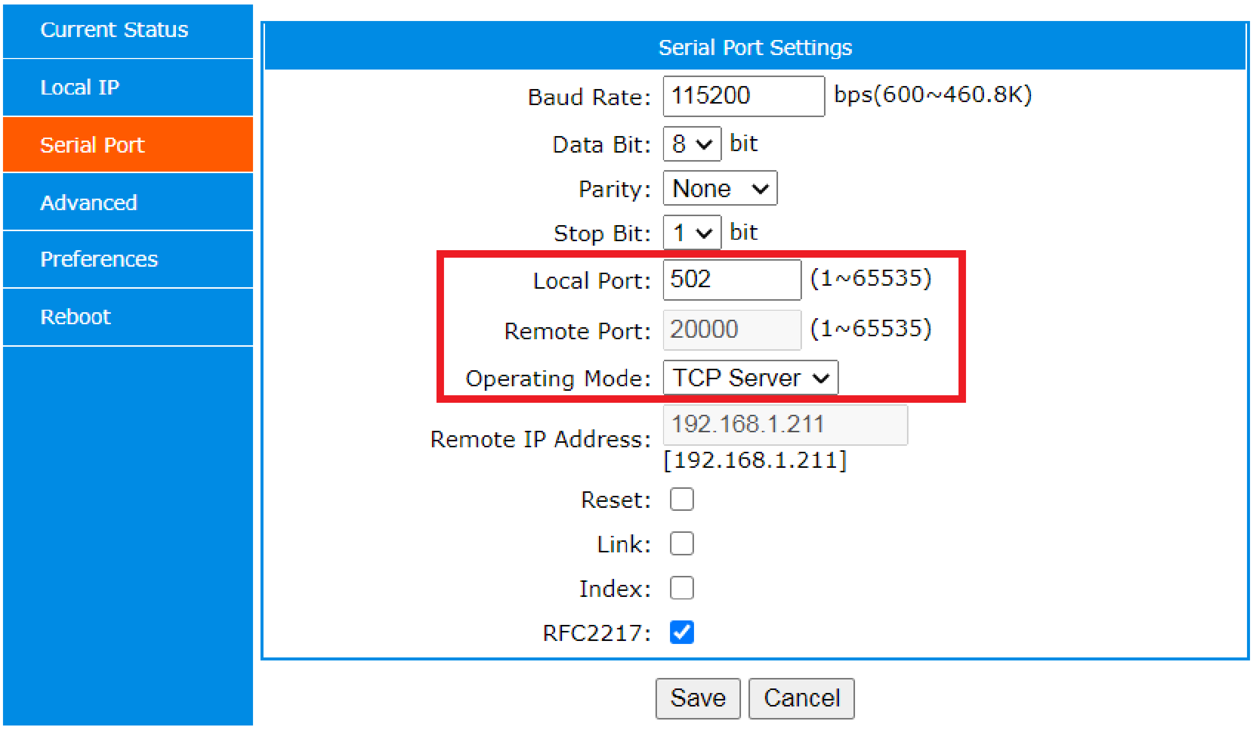
Local IP Settings

IP Address Type:	Static IP			
DNS type:	Auto			
Static IP:	192	168	1	201
Subnet Mask:	255	255	255	0
Gateway:	192	168	1	254
DNS Server:	8	8	8	8

3) Go to "Serial Port" and choose the correct COM port settings. Note: The COM port settings must match with the RS-485 device to which the DVS-485-2 is connected.

4) Set the device server as TCP Server and port number as 502. The device server becomes a TCP Sever now.

Note: For more detailed information about Web Management tools, please refer to "Web Management".



Serial Port Settings

Baud Rate: 115200 bps(600~460.8K)

Data Bit: 8 bit

Parity: None

Stop Bit: 1 bit

Local Port: 502 (1~65535)

Remote Port: 20000 (1~65535)

Operating Mode: TCP Server

Remote IP Address: 192.168.1.211 [192.168.1.211]

Reset: □

Link: □

Index: □

RFC2217: ☑

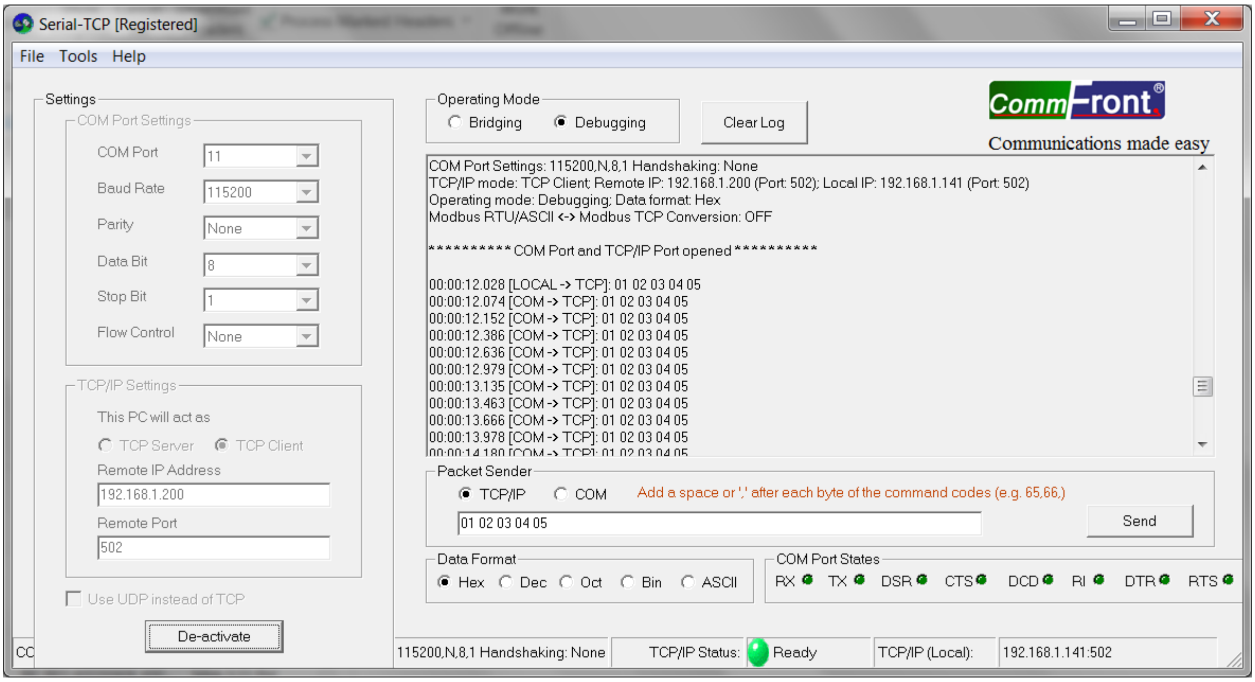
4.2.5 Test with the Serial-TCP software

The Serial-TCP can perform as a software-type device server so users can test out most of the functions with TCP server, TCP client, UDP server, and UDP client operating modes. In addition, the Serial-TCP comes with a "Packet Sender" tool so users can send commands directly to device server's COM port or TCP/UDP port, which is very useful for testing the device server.

1) Download and install the free version of CommFront's Serial-TCP software (download link: https://www.commfront.com/pages/downloads).

2) Set Serial-TCP as a TCP client and connect to device server's TCP port (assume that the device server is set as a TCP server).

3) Send HEX command "01 02 03 04 05" to TCP port. When the command reaches the device server's TCP port, it will be forwarded to device server's COM port and come back to Serial-TCP software's COM port, thus creating an infinite loop. Click the "De-activate" button to stop the loop.



Settings

COM Port Settings

COM Port: 11

Baud Rate: 115200

Parity: None

Data Bit: 8

Stop Bit: 1

Flow Control: None

TCP/IP Settings

This PC will act as ○ TCP Server ● TCP Client

Remote IP Address: 192.168.1.200

Remote Port: 502

□ Use UDP instead of TCP

Operating Mode: ○ Bridging ● Debugging

```
COM Port Settings: 115200,N,8,1 Handshaking: None
TCP/IP mode: TCP Client; Remote IP: 192.168.1.200 (Port: 502); Local IP: 192.168.1.141 (Port: 502)
Operating mode: Debugging; Data format: Hex
Modbus RTU/ASCII <-> Modbus TCP Conversion: OFF

********** COM Port and TCP/IP Port opened **********

00:00:12.028 [LOCAL -> TCP]: 01 02 03 04 05
00:00:12.074 [COM -> TCP]: 01 02 03 04 05
00:00:12.152 [COM -> TCP]: 01 02 03 04 05
00:00:12.386 [COM -> TCP]: 01 02 03 04 05
00:00:12.636 [COM -> TCP]: 01 02 03 04 05
00:00:12.979 [COM -> TCP]: 01 02 03 04 05
00:00:13.135 [COM -> TCP]: 01 02 03 04 05
00:00:13.463 [COM -> TCP]: 01 02 03 04 05
00:00:13.666 [COM -> TCP]: 01 02 03 04 05
00:00:13.978 [COM -> TCP]: 01 02 03 04 05
00:00:14.180 [COM -> TCP]: 01 02 03 04 05
```

Packet Sender: ● TCP/IP ○ COM

Add a space or ',' after each byte of the command codes (e.g. 65,66,)

01 02 03 04 05

Data Format: ● Hex ○ Dec ○ Oct ○ Bin ○ ASCII

COM Port States: RX ● TX ● DSR ● CTS ● DCD ● RI ● DTR ● RTS ●

CC | 115200,N,8,1 Handshaking: None | TCP/IP Status: ● Ready | TCP/IP (Local): | 192.168.1.141:502

4.2.6 Test with the 232Analyzer software

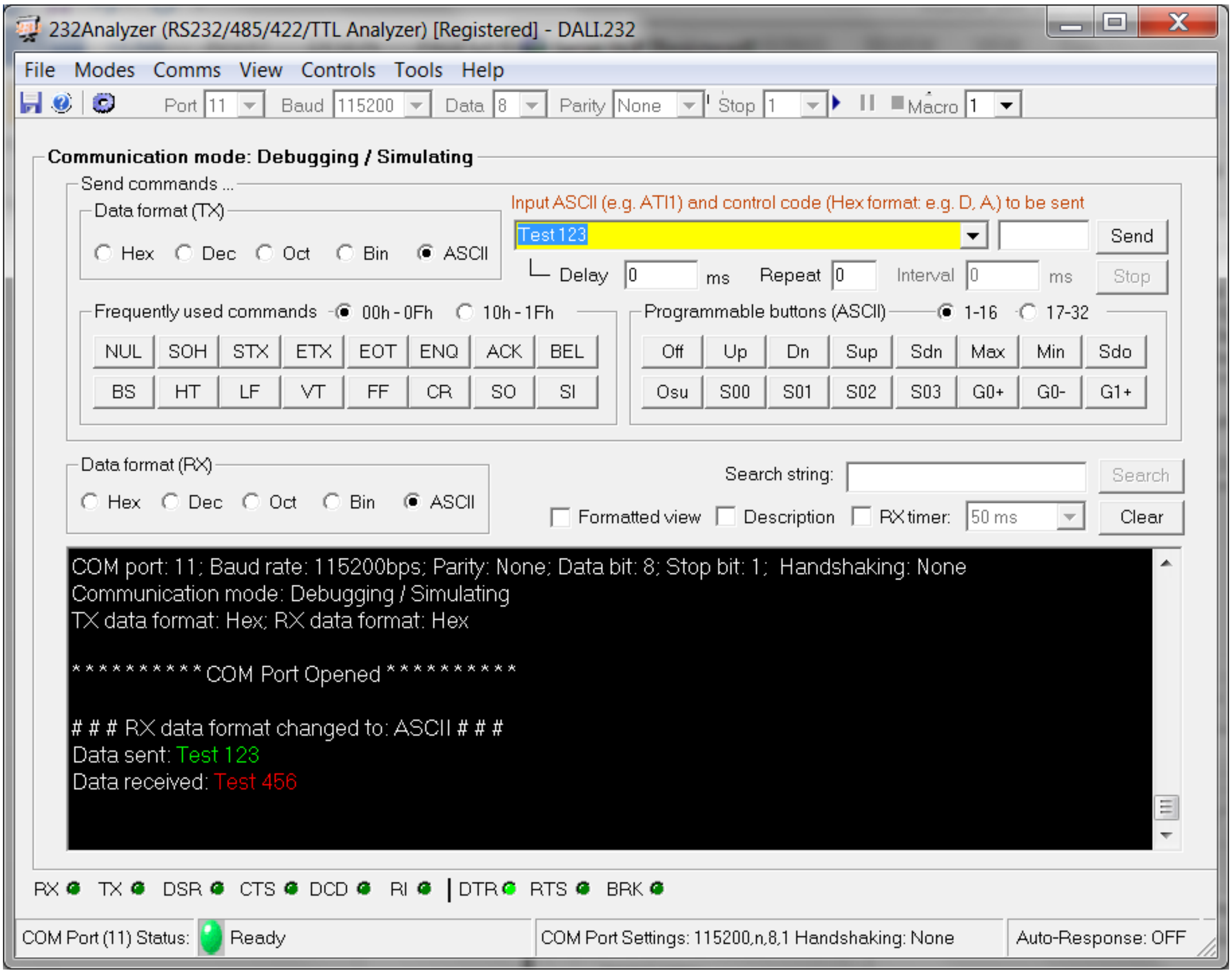
The 232Analyzer software can act as a Serial Terminal so you can send commands directly to device server's COM port. In addition, you can connect Serial-TCP software (which acts as a TCP client for the example below) to the device server. When the commands are sent from the 232Analyzer software, the Serial-TCP should receive the same. Similarly, when commands are sent from the Serial-TCP software, the 232Analyzer software should receive the same commands.

1) Download and install the free version of CommFront's 232Analyzer software (download link: https://www.commfront.com/pages/downloads).

2) Set Serial-TCP as a TCP client and connect to the device server's TCP port (assume that the device server is set as a TCP server).

3) Send ASCII command "Test 123" to TCP port. When the command reaches the device server's TCP port, it will be forwarded to device server's COM port and then received by the 232Analyzer software.

4) Send ASCII command "Test 456" from the 232Analyzer software, and you should receive the same on the Serial-TCP software.


232Analyzer (RS232/485/422/TTL Analyzer) [Registered] - DALI.232
File Modes Comms View Controls Tools Help
Send commands ...
Data format (TX)
Frequently used commands
NUL SOH STX ETX EOT ENQ ACK BEL
BS HT LF VT FF CR SO SI
Programmable buttons (ASCII)
Off Up Dn Sup Sdn Max Min Sdo
Osu S00 S01 S02 S03 G0+ G0- G1+
Send
Stop
Search
Clear
RX TX DSR CTS DCD RI DTR RTS BRK

Port 11 Baud 115200 Data 8 Parity None Stop 1 Macro 1

Communication mode: Debugging / Simulating

Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent: Test 123

Delay 0 ms Repeat 0 Interval 0 ms

Frequently used commands ● 00h - 0Fh ○ 10h - 1Fh

Programmable buttons (ASCII) ● 1-16 ○ 17-32

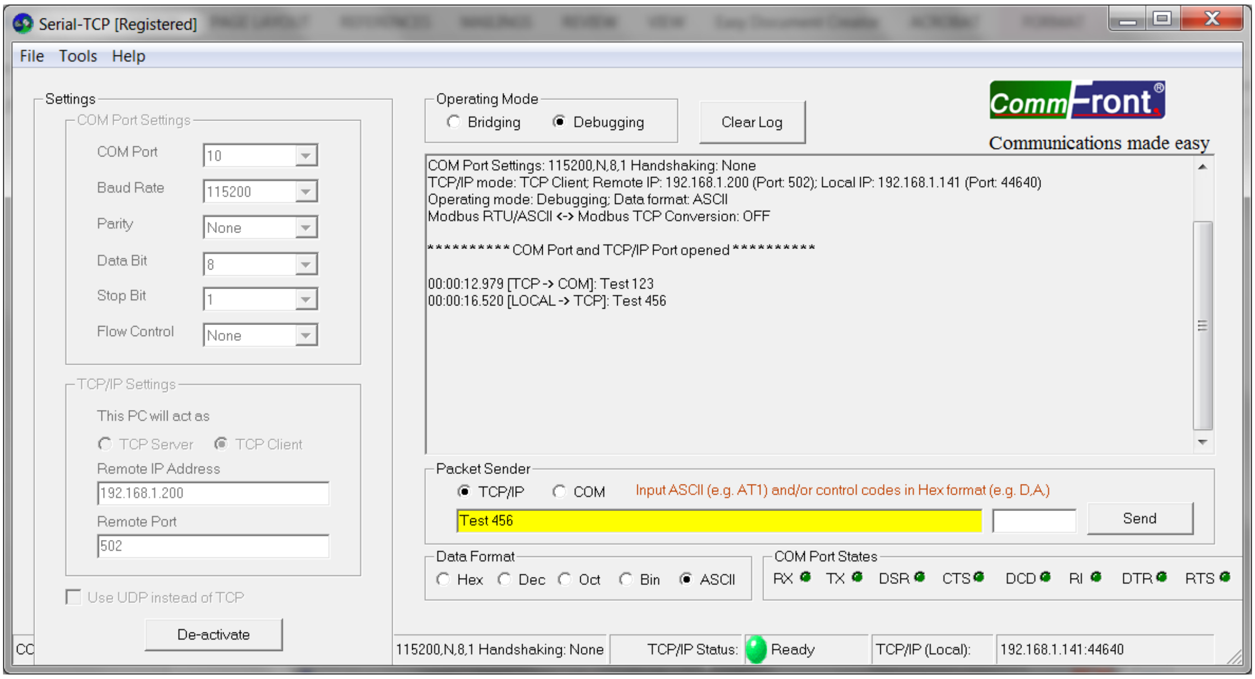
Data format (RX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Search string:

□ Formatted view □ Description □ RX timer: 50 ms

```
COM port: 11; Baud rate: 115200bps; Parity: None; Data bit: 8; Stop bit: 1;  Handshaking: None
Communication mode: Debugging / Simulating
TX data format: Hex; RX data format: Hex

********** COM Port Opened **********

# # # RX data format changed to: ASCII # # #
Data sent: Test 123
Data received: Test 456
```

COM Port (11) Status: Ready | COM Port Settings: 115200,n,8,1 Handshaking: None | Auto-Response: OFF



Settings

COM Port Settings

COM Port: 10
Baud Rate: 115200
Parity: None
Data Bit: 8
Stop Bit: 1
Flow Control: None

TCP/IP Settings

This PC will act as ○ TCP Server ● TCP Client

Remote IP Address: 192.168.1.200

Remote Port: 502

□ Use UDP instead of TCP

Operating Mode: ○ Bridging ● Debugging

```
COM Port Settings: 115200,N,8,1 Handshaking: None
TCP/IP mode: TCP Client; Remote IP: 192.168.1.200 (Port: 502); Local IP: 192.168.1.141 (Port: 44640)
Operating mode: Debugging; Data format: ASCII
Modbus RTU/ASCII <-> Modbus TCP Conversion: OFF

********** COM Port and TCP/IP Port opened **********

00:00:12.979 [TCP -> COM]: Test 123
00:00:16.520 [LOCAL -> TCP]: Test 456
```

Packet Sender: ● TCP/IP ○ COM

Input ASCII (e.g. AT1) and/or control codes in Hex format (e.g. D,A,): Test 456

Data Format: ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

COM Port States

115200,N,8,1 Handshaking: None | TCP/IP Status: Ready | TCP/IP (Local): 192.168.1.141:44640

APPENDIX: AT COMMANDS SET

1. Introduction

AT commands are used for controlling modem, GSM, and many other types of communication equipment that come with a serial port. CommFront's device servers use AT commands for querying or setting parameters such as COM port and IP parameters.

2. Syntax

"AT" stands for "attention", and it must be used at the beginning of each command line. The AT command must be terminated with a <CR> code (HEX code: 0D). AT commands are usually responded to by the device, and the response looks something like: "AT+COMMAND<CR><CR><LF><**response**><CR><LF>". Throughout this document, only the responses are presented. "<CR><LF>" is omitted intentionally.

Command syntax

Query command:
AT+COMMAND<CR>

Set command:
AT+COMMAND=<parameter1,parameter2,…><CR>

Response syntax

Query command:
+OK=<parameter1,parameter2,…>

Set command:
+OK

Error:
+ERR=<number>

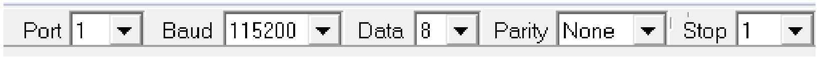
Error codes

Error code	Description
1	Invalid format
2	Invalid command
3	Invalid operator
4	Invalid parameter
5	Invalid operation

3. Software Tools and Examples

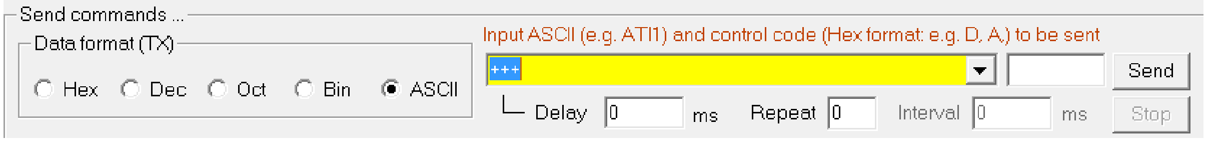
1) CommFront's 232Analyzer software provides an easy and fast way for sending AT commands (download link: https://www.commfront.com/pages/downloads).

2) Run the 232Analyzer software and set the COM port settings to 115.2Kbps, 8, N, 1. Then select "ASCII" for both TX and RX data formats.



Port 1 Baud 115200 Data 8 Parity None Stop 1

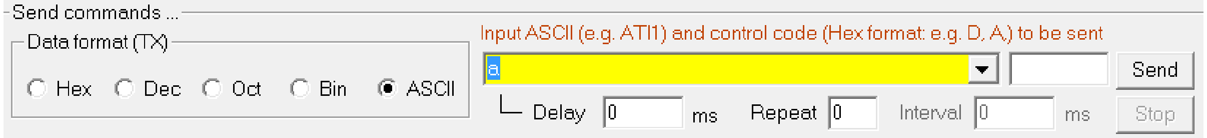
3) To enter AT Command mode, first send the ASCII code: **+++**



Send commands ...
Data format (TX) ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
+++
Delay 0 ms Repeat 0 Interval 0 ms

4) The device server responds with an ASCII code: **a**

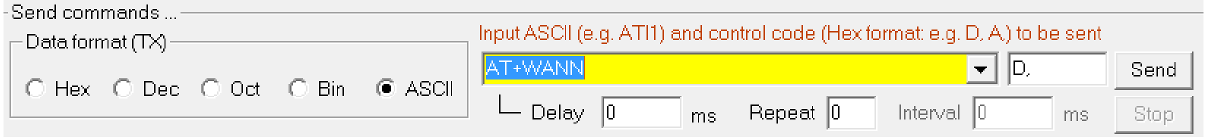
5) Send the second ASCII code within 3 seconds: **a**



Send commands ...
Data format (TX) ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
a
Delay 0 ms Repeat 0 Interval 0 ms

6) The device server responds with an ASCII code and it now enters into "AT Command" mode: **+ok**

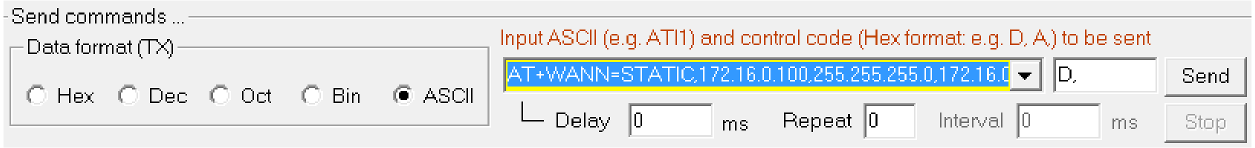
7) Query current network settings: **AT+WANN<CR>**



Send commands ...
Data format (TX) ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
AT+WANN D,
Delay 0 ms Repeat 0 Interval 0 ms

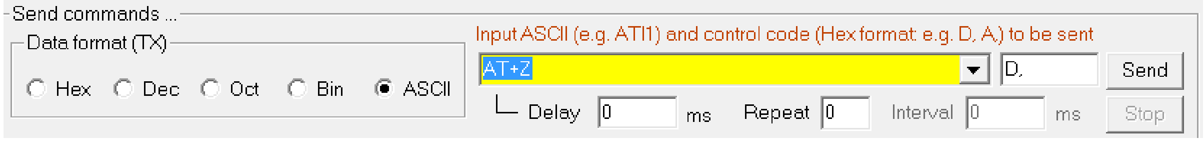
8) The device server responds with the current network settings: **+OK=STATIC,192.168.1.201,255.255.255.0,192.168.1.254**

9) Change current network settings to the same sub network as your PC's, e.g.: **AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0.1 <CR>**



Send commands ...
Data format (TX) ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII
Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent
AT+WANN=STATIC,172.16.0.100,255.255.255.0,172.16.0 D,
Delay 0 ms Repeat 0 Interval 0 ms

10) The device server responds with a confirmation code: **+OK**
11) Restart the device server for the new settings to take effect: **AT+Z<CR>**



Data format (TX): ○ Hex ○ Dec ○ Oct ○ Bin ● ASCII

Input ASCII (e.g. ATI1) and control code (Hex format: e.g. D, A,) to be sent: AT+Z | D,

Delay 0 ms Repeat 0 Interval 0 ms

12) The device server restarts and exits the "AT command" mode.

4. AT Commands Set

1) Switch from data mode to AT command mode

Command	Response
+++	**a**
Send next command within 3 seconds	
a	**+ok**
	Enter AT Command mode

2) Switch from AT command mode to data mode

Command	Response
AT+ENTM<CR>	**+OK**
	Exit AT Command mode

3) Restart the device

Command	Response
AT+Z<CR>	**+OK**
	Restart the device and exit AT Command mode

4) Restore factory settings

Command	Response
AT+RELD<CR>	**+OK**
	Require device restart for factory settings to take effect

5) Query MAC address

Command	Response
AT+MAC<CR>	**+OK=<MAC>**
	Example: +OK=D8C561010001

6) Query or set Network parameters

Command (type=DHCP/STATIC)	Response
AT+WANN<CR>	**+OK=<type,IP_address,subnet_mask,gateway>**
AT+WANN=<type,IP_address,subnet_mask,gateway><CR>	**+OK**
Example: AT+WANN=STATIC, 192.168.1.201,255.255.255.0,192.168.1.254<CR>	

7) Query or set DNS address

Command	Response
AT+DNS<CR>	**+OK=<address>**
AT+DNS=<address><CR>	**+OK**
Example: AT+DNS=8.8.8.8<CR>	

8) Query or set COM port parameters

Command	Response
AT+UART<CR>	**+OK=<baud_rate,data_bit,stop_ bit,parity,flowctrl>**
AT+UART=<baud_rate,data_bit,stop_ bit,parity,flow_ctrl><CR>	**+OK**
Baud rate: 9600,19200,38400,57600, 115200,128000,256000 Data bit: 7,8 Stop bit: 1,2 Parity: None, Even, Odd, Mask, Space Flow Ctrl: No Hardware Control (NFC), Hardware Control (FCH)	
Example: AT+UART=115200,8,1,NONE,NFC<CR>	

9) Query or set Socket parameters

Command	Response
AT+SOCK<CR>	**+OK=<socket,IP,port>**
AT+SOCK=<socket,IP,port><CR>	**+OK**
Socket: TCPS – TCP Server TCPC – TCP Client UDPS – UDP Server UDPC – UDP Client HTPC – Httpd Client IP: Remote IP address or domain Port: Local port (for server) or Remote port (for client)	
Example 1: AT+SOCK=TCPS, 192.168.1.211,502<CR> Example 2: AT+SOCK=TCPC, 192.168.1.211,20000<CR>	

10) Query TCP connection status

Command	Response
AT+SOCKLK<CR>	**+OK=<status>**
Status: CONNECT – TCP connection has been established DISCONNECT – TCP connection has not been established	
	Example: +OK=CONNECT

11) Query or set Local port

Command	Response
AT+SOCKPORT<CR>	**+OK=<number>**
AT+SOCKPORT=<number><CR>	**+OK**
Number: 0 – Random 1-65535 – Local port	
Example: AT+SOCKPORT=502<CR>	

12) Query Production time

Command	Response
AT+PDTIME<CR>	**+OK=<time>**
	Example: +OK=2021-01-01 15:43:11

13) Query or set Registry packet type

Command	Response
AT+REGEN<CR>	**+OK=<type>**
AT+REGEN=<type><CR>	**+OK**
Type: OFF: No registry packet is used MAC – Use MAC address as registry packet USR – Use user-defined packet CLOUD – Use cloud packet	
Example: AT+REGEN=MAC<CR>	

14) Query or set When to send registry packet

Command	Response
AT+REGTCP<CR>	**+OK=<type>**
AT+REGTCP=<type><CR>	**+OK**
Type: FIRST – Send together with connection string EVERY – Send together with data string ALL – Send together with connection and data string	
Example: AT+REGTCP=FIRST<CR>	

15) Query or set Device ID and cloud key for cloud connection

Command	Response
AT+REGCLOUD<CR>	**+OK=<deviceID,cloudKey>**
AT+REGCLOUD=<deviceID,cloudKey><CR>	**+OK**
Device ID – Device ID for cloud connection Cloud Key – Cloud key for cloud connection	
Example: AT+REGCLOUD=00000001,0001<CR>	

16) Query or set User-defined registry packet

Command	Response
AT+REGUSR<CR>	**+OK=<data>**
AT+REGUSR=<data><CR>	**+OK**
Data: ASCII codes (up to 40 bytes)	
Example: AT+REGUSR=www.commfront.com <CR>	

17) Query or set HTTP request method

Command	Response
AT+HTPTP<CR>	**+OK=<method>**
AT+HTPTP=<method><CR>	**+OK**
Method: GET – Use GET method for HTTP request POST – Use POST method for HTTP request	
Example: AT+HTPTP=GET<CR>	

18) Query or set HTTP request URL

Command	Response
AT+HTPURL<CR>	**+OK=<url>**
AT+HTPURL=<url><CR>	**+OK**
URL: Less than 100 characters	
Example: AT+HTPURL=/1.php<CR>	

19) Query or set HTTP request header

Command	Response
AT+HTPHEAD<CR>	**+OK=<data>**
AT+HTPHEAD=<data><CR>	**+OK**
Header: Less than 200 characters	
Example: AT+HTPHEAD=User_Agent: Mozila/4.0<CR>	

20) Enable or disable Remove HTTP request header

Command	Response
AT+HTPCHD<CR>	**+OK=<status>**
AT+HTPCHD=<status><CR>	**+OK**
Status: ON – Remove HTTP request header OFF – Keep HTTP request header	
Example: AT+HTPCHD=OFF<CR>	

21) Enable or disable Heartbeat function

Command	Response
AT+HEARTEN<CR>	**+OK=<status>**
AT+HEARTEN=<status><CR>	**+OK**
Status: ON – Enable heartbeat function OFF – Disable heartbeat function	
Example: AT+HEARTEN=ON<CR>	

22) Query or set Direction to send heartbeat packet

Command	Response
AT+HEARTTP<CR>	**+OK=<type>**
AT+HEARTTP=<type><CR>	**+OK**
Type: COM – Send heartbeat packet to COM port NET – Send heartbeat packet to network port	
Example: AT+HEARTTP=NET<CR>	

23) Query or set Heartbeat interval

Command	Response
AT+HEARTTM<CR>	**+OK=<time>**
AT+HEARTTM=<time><CR>	**+OK**
Time: 1 – 65535s	
Example: AT+HEARTTM=30<CR>	

24) Query or set Heartbeat packet

Command	Response
AT+HEARTDT<CR>	**+OK=<data>**
AT+HEARTDT=<data><CR>	**+OK**
Data: ASCII codes (up to 40 bytes)	
Example: AT+HEARTDT=www.commfront.com<CR>	

25) Enable or disable Restart after 30 times of unsuccessful connection attempts (TCP client mode)

Command	Response
AT+CLIENTRST<CR>	**+OK=<status>**
AT+CLIENTRST=<status><CR>	**+OK**
Status: ON – Enable restart after 30 times of unsuccessful TCP connection attempts OFF – Disable restart after 30 times of unsuccessful TCP connection attempts	
Example: AT+CLIENTRST=ON<CR>	

26) Enable or disable Link function

Command	Response
AT+SCSLINK<CR>	**+OK=<status>**
AT+SCSLINK=<status><CR>	**+OK**
Status: ON – Socket link function is ON OFF – Socket link function is OFF	
Example: AT+SCSLINK=ON<CR>	

27) Enable or disable Index function (TCP server mode)

Command	Response
AT+INDEXEN<CR>	**+OK=<status>**
AT+INDEXEN=<status><CR>	**+OK**
Status: ON – Enable index function OFF – Disable index function	
Example: AT+INDEXEN=ON<CR>	

28) Enable or disable RFC2217

Command	Response
AT+RFCEN<CR>	**+OK=<status>**
AT+RFCEN=<status><CR>	**+OK**
Status: ON – Enable RFC2217 OFF – Disable RFC2217	
Example: AT+RFCEN=ON<CR>	

29) Enable or disable Short connection function

Command	Response
AT+SOCKSL<CR>	**+OK=<status>**
AT+SOCKSL=<status><CR>	**+OK**
Status: ON – Enable short connection function OFF – Disable short connection function	
Example: AT+SOCKSL=ON<CR>	

30) Query or set Short connection timeout

Command	Response
AT+SHORTO<CR>	**+OK=<time>**
AT+SHORTO=<time><CR>	**+OK**
Time: 2-255s	
Example: AT+SHORTO=3<CR>	

31) Query or set Server actions when maximum number of client connections is reached

Command	Response
AT+TCPSE<CR>	**+OK=<action>**
AT+TCPSE=<action><CR>	**+OK**
Status: KEEP – Keep old connection KICK – Disable old connection and allow new connection	
Example: AT+TCPSE=KEEP<CR>	

32) Enable or disable Clear serial buffer upon TCP/IP connection

Command	Response
AT+UARTCLBUF<CR>	**+OK=<status>**
AT+UARTCLBUF=<status><CR>	**+OK**
Status: ON – Enable Clear serial buffer upon TCP/IP connection OFF – Disable Clear serial buffer upon TCP/IP connection	
Example: AT+UARTCLBUF=ON<CR>	

33) Enable or disable Modbus RTU to Modbus TCP conversion

Command	Response
AT+MODTCP<CR>	**+OK=<status>**
AT+MODTCP=<status><CR>	**+OK**
Status: ON – Enable Modbus RTU to Modbus TCP conversion OFF – Disable Modbus RTU to Modbus TCP conversion	
Example: AT+MODTCP=ON<CR>	

34) Query or set Device name

Command	Response
AT+MID<CR>	**+OK=<name>**
AT+MID=<name><CR>	**+OK**
Name: Up to 14 characters	
Example: AT+MID=DVS-485-2<CR>	

35) Query or set Web port

Command	Response
AT+WEBPORT<CR>	**+OK=<port>**
AT+WEBPORT=<port><CR>	**+OK**
Example: AT+WEBPORT=80<CR>	

36) Query or set User_name and password

Command	Response
AT+WEBU<CR>	**+OK=<username,password>**
AT+WEBU=<username,password> <CR>	**+OK**
Example: AT+WEBU=user,password<CR>	

37) Query Firmware version

Command	Response
AT+VER<CR>	**+OK=<ver>**

38) Query or set Maximum number or client connections (server mode)

Command	Response
AT+MAXSK<CR>	**+OK=<number>**
AT+MAXSK=<number><CR>	**+OK**
Number: 1 to 8	
Example: AT+MAXSK=8<CR>	

39) Query or set No data auto-restart timeout

Command	Response
AT+RSTIM<CR>	**+OK=<time>**
AT+RSTIM=<time><CR>	**+OK**
Time: 0 – No auto-restart 60-65535s – Auto-restart timeout if no data received from serial or network port	
Example: AT+RSTIM=3600<CR>	

40) Get Help information

Command	Response
AT+H<CR>	**+OK=<info>**