User Manual for

USC520:
Serial Ethernet Converter (RS232, RS485, RS422)
Industrial Panel Mount, 2 Ports

USC540:
Serial Ethernet Converter (RS232, RS485, RS422)
Industrial Panel Mount, 4 Ports
Contents

2. Overview ......................................................................................................................................................... 4
   2.1. Brief Introduction .................................................................................................................................. 5
   2.2. Features ............................................................................................................................................... 5
   2.3. Basic Parameter ................................................................................................................................... 6
   2.4. Dimension ............................................................................................................................................ 6
   2.5. Indicators and DIP switches ................................................................................................................... 7
   2.6. DB9 Pin Definition ................................................................................................................................ 9
3. Product Function .......................................................................................................................................... 10
   3.1. Socket A Communication ................................................................................................................... 11
      3.1.1. TCP Client Model ..................................................................................................................... 11
      3.1.2. TCP Server Mode ..................................................................................................................... 12
      3.1.3. UDP Client Mode ..................................................................................................................... 12
      3.1.4. UDP Server Mode ..................................................................................................................... 12
      3.1.5. TCP and UDP Comparison ...................................................................................................... 13
      3.1.6. HTTPD Client ........................................................................................................................... 13
   3.2. Socket B Communication ................................................................................................................... 14
   3.3. Modbus Gateway ................................................................................................................................ 16
      3.4.1. Transmit modbus protocol in transparent mode ...................................................................... 16
      3.4.2. Modbus RTU to Modbus TCP .................................................................................................. 17
      3.4.3. Modbus Active Query Function .............................................................................................. 18
      3.4.4. Modbus Polling Function ......................................................................................................... 20
   3.4. Extra functions ................................................................................................................................... 23
      3.5.1. DHCP ....................................................................................................................................... 23
      3.5.2. DNS ......................................................................................................................................... 23
      3.5.3. Self-Defined Heartbeat Function ............................................................................................. 24
      3.5.4. Self-Defined Registration Package Function ........................................................................... 25
      3.5.5. Web to serial ............................................................................................................................ 27
      3.5.6. Network Printing Function ....................................................................................................... 28
      3.5.7. Serial Port Packaging Mechanism .......................................................................................... 30
      3.5.8. Flow Calculation ...................................................................................................................... 30
      3.5.9. Synchronous baud rate (RFC2217) ........................................................................................... 31
      3.5.10. KeepAlive ............................................................................................................................... 32
      3.5.11. Device ID ................................................................................................................................ 32
      3.5.12. Webpage Port ........................................................................................................................... 32
      3.5.13. Revise MAC ............................................................................................................................ 32
      3.5.14. Firmware Upgrade .................................................................................................................. 33
      3.5.15. Hardware Flow Control RTS/CTS & XON/XoFF ................................................................... 33
3.5.17. Reset ...................................................................................................................................... 33

4. Setting Protocol ............................................................................................................................................ 33
   4.1. Network Setting Protocol ................................................................................................................... 33
      4.1.1. Set Parameter Process ........................................................................................................... 33
      4.1.2. Setting Command Content ...................................................................................................... 34
      4.1.3. Commands’ Return Content .................................................................................................... 38
   4.2. Serial Setting Protocol ....................................................................................................................... 39
      4.2.1. AT Command ........................................................................................................................... 39
      4.2.2. Enter AT Command Mode ....................................................................................................... 41
      4.2.3. AT Command details ............................................................................................................... 41

5. Parameter Configuration .............................................................................................................................. 45
   5.1 Software Configuration ....................................................................................................................... 45
### 1.1. Default Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>admin</td>
</tr>
<tr>
<td>Password</td>
<td>admin</td>
</tr>
<tr>
<td>IP address</td>
<td>192.168.0.7</td>
</tr>
<tr>
<td>Mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Gateway</td>
<td>192.168.0.1</td>
</tr>
<tr>
<td>Work Mode of Port 1</td>
<td>TCP Server</td>
</tr>
<tr>
<td>Local Port of Port 1</td>
<td>23</td>
</tr>
<tr>
<td>Work Mode of Port 2</td>
<td>TCP Server</td>
</tr>
<tr>
<td>Local Port of Port 2</td>
<td>26</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>115200</td>
</tr>
<tr>
<td>Parity bit/Data bit/Stop bit</td>
<td>None/8/1</td>
</tr>
</tbody>
</table>
2. Overview

2.1. Brief Introduction

USC520/USC540 is used to transmit data transparently between TCP/UDP data packets and a RS232/RS485/RS422 interface. It has an ARM processor, low power, fast speed, high stability and two/four serial ports which can work as RS232, RS485 or RS422 serial interface which compiles to industrial standard.

2.2. Features

1. ARM base on Cortex-M4 kernel, and reliable TCP/IP protocol stack
2. Industrial working temperature range from -40C ~85C
3. Auto-MDI/MDIX,RJ45 port with 10/100Mbps
4. Support TCP server, TCP client, UDP, UDP server and HTTPD client work mode
5. One port corresponding to two socket
6. Support network printing via IP address
7. Function of Modbus gateway, modbus RTU to modbus TCP, modbus multi-host Polling
8. Two serial port, each port can work as RS232 or RS485 or RS422 and work individually
9. Distinguish which serial port connect to device via port number
10. Support virtual serial port and provide corresponding software USR-VCOM
11. Serial baud rate from 600bps to 230.4K bps; Check bit of None,Odd,Even,Mark and Space
12. Support static IP, DHCP/DNS and search devices within network through UDP broadcast.
13. Provide serial and network setting protocol, TCP/IP socket example code such as VB, C++, Delphi, Android, IOS
14. Built-in web page; Customized web page is acceptable
15. Reload button, one key to restore default settings
16. RJ45 with Link/Data indicator light, built-in isolation transformer and 2 KV electromagnetic isolation
17. The global unique MAC address bought from IEEE, also user can define MAC address
18. Upgrade firmware via network
19. Support web port revise (80 by default)
20. Keepalive, detect dead links and reconnect rapidly
21. Support account and password, used to page log in and network settings safely
22. Support one channel Web socket, realize bidirectional transparent transmission between web page and serial side
23. Power supply in two mode, DC adapter or 5.08-2 terminal pin
24. Communication indicator light of two serial port: RX/TX
25. UPD broadcast function, can receive/send data to all IP in the same network

2.3. Basic Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage</td>
<td>DC5~36V</td>
</tr>
<tr>
<td>Working Current</td>
<td>90mA@5V</td>
</tr>
<tr>
<td>Operating Temp.</td>
<td>-40~+85°C</td>
</tr>
<tr>
<td>Power</td>
<td>&lt;1W</td>
</tr>
<tr>
<td>Storage Temp.</td>
<td>-45<del>105°C, 5</del>95%RH</td>
</tr>
</tbody>
</table>

2.4. Dimensions
2.5. Indicators and DIP switches

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>① Power</td>
<td>Indicate power. On when power is supplied</td>
</tr>
<tr>
<td>② Work</td>
<td>Indicate working status. It blinks when N520/N540 works well. If it is on or off for a period, N520/N540 works improperly, you should power cycle to restart.</td>
</tr>
<tr>
<td>③ TX1</td>
<td>Blinks when port 1 sends data</td>
</tr>
<tr>
<td>④ RX1</td>
<td>Blinks when port 1 receive data</td>
</tr>
<tr>
<td>⑤ TX2</td>
<td>Blinks when port 2 sends data</td>
</tr>
<tr>
<td>⑥ RX2</td>
<td>Blinks when port 2 receive data</td>
</tr>
</tbody>
</table>
- For RS232 port, DIP switches are down, such as the below picture.

- For RS485 port, DIP switches are up, such as the below picture.

- For RS422 port, DIP switches on the left are up, DIP switch on the light are down,
2.6. DB9 Pin Definition

For RS232 port, pin definition is as below

<table>
<thead>
<tr>
<th>Pin</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RX, pin of receives data</td>
</tr>
<tr>
<td>3</td>
<td>TX</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
</tr>
</tbody>
</table>

For RS485 port, pin 3 works as “B(−)”, Pin 7 works as “A+”

For RS422 port

<table>
<thead>
<tr>
<th>Pin</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RX+, pin of receives data</td>
</tr>
<tr>
<td>3</td>
<td>RX-</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>TX+</td>
</tr>
<tr>
<td>8</td>
<td>TX-</td>
</tr>
</tbody>
</table>
3. Product Function

The following is function diagram of USC520/USC540

Each serial port corresponds to two sockets: socket A and socket B. Socket B can be enabled or disabled.
3.1. Socket A Communication

3.1.1. TCP Client Model

4.1.1 TCP Client Model

1) Different from UDP mode, in this mode, connection has status of ‘disconnected’ and active. Connection is still active although USC520/USC540 does not send data.

2) Identify disconnects. After connection built, it sends keepalive searching packet every 15 seconds. Once there is an interrupt, it can be detected rapidly then make USC520/USC540 disconnect from former connection and reconnect.

3) It will connect to same source port when USC520/USC540 try to connect server and local port is not “0”.

4) It supports Synchronous baud rate (Similar RCF2217), which can revise USC520/USC540’s serial parameter as baud rate accordingly. This function should be combined with USR-VCOM.

5) Under the same LAN, USC520/USC540 must be in the same network segment then can communicate. If not, USC520/USC540 must be set with right one.

6) Support USR Cloud.

7) Support Modbus TCP function.

8) USC520/USC540 work as TCP Client, it connects to TCP server, Destination IP and port should be cared. The IP can be device with same LAN, also can be different LAN or cross public network. If it connects to server cross public network, the server should have public IP.

9) USC520/USC540 work under TCP Client, It connects to the target IP/Port automatically, will not accept other connection request.

10) As TCP Client, need to set USR-N510’s local port number to be “0” then it can visit server with randomized port number, so that it can solve unsuccessful re-connection in case server judge connection status abnormally and shield USC520/USC540 re-connection request.
3.1.2. TCP Server Mode

1) Different from UDP, in this mode, connection has status of disconnected and active. Connection is still active although USC520/USC540 does not send data.
2) USC520/USC540 listens to local port set firstly, respond and build connection when there is a connection request. Serial port will send data to all client which connected with USC520/USC540 at the same time once serial port received data.
3) It supports USR Synchronous baud rate (Similar RCF2217), which can revise USC520/USC540 serial parameter as baud rate accordingly. This function should be combined with USR-VCOM.
4) It support 8 clients connections at max. (32 clients will be improved in the following.)
5) Support Modbus TCP function.
6) Under TCP Server mode, USC520/USC540 listens to local port actively and will not monitor connected IP and port. When the 9th client is connected, the oldest one will be ticked.

3.1.3. UDP Client Mode

1) It subjects to UDP protocol, no connection, just only sending data.
2) USC520/USC540 only communicates with destination port of IP. Otherwise, the data cannot be received.
3) Under this mode, destination Address is 255.255.255.255, then it can make UDP broadcast and receive broadcast data. Broadcast within segment as 192.168.0.255, it can be sent but cannot be received currently.
4) Under UDP Client/UDP Server mode, host PC allow data length 1460 bytes at max to USC520/USC540.

3.1.4. UDP Server Mode

1) Based on normal UDP, it doesn't verify source IP address. Every time USC520/USC540 receives one UDP data packet, it changes destination IP to where data comes and it sends the data to the IP and port which communicate latest.
3.1.5. TCP and UDP Comparison

<table>
<thead>
<tr>
<th></th>
<th>TCP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advantages</td>
<td>Stable, no loss</td>
<td>No Connection mechanism, simple, flexible</td>
</tr>
<tr>
<td></td>
<td>Reliable connection mechanism</td>
<td>Suit for small packet and high frequency</td>
</tr>
<tr>
<td></td>
<td>Resend after data sending fails</td>
<td>Accurate data sending interval</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>Long packet starting</td>
<td>More less under bad network environment</td>
</tr>
<tr>
<td></td>
<td>Jam for small packet and high frequency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inaccurate interval resulted from check</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and resend mechanism</td>
<td></td>
</tr>
</tbody>
</table>

3.1.6. HTTPD Client

It is used to transmit data from USC520/USC540 to HTTP server or gain data from HTTP server. USC520/USC540 can handle complex HTTP protocol so user just do programming for serial, and not need to worry about HTTP.

When USC520/USC540 sends data to HTTP server via serial port, it only needs to send the header of requested data; All the returned data will be transmitted by USC520/USC540, user need to analyze the packets.

Test Example:
1) Entry [http://192.168.0.7](http://192.168.0.7) (N520’s IP) to open its web page
   1. Set USC520/USC540 as HTTPD Client.
   2. Set HTTPD packet Header.

   <Note>:
   - HTTPD Client only support GET to request HTTPD Server. POST will be available in the following.
   - GET/ is fixed packet header.
   - 1.php?data= is the visited/submitted the page
   - $ stands for data sent by serial (Serial port does not need to send “$”)
   - HTTP/1.1 is requested protocol.
   - Host is means requested IP address/ domain.
   - Enter twice

2) Save the parameters and restart USC520/USC540.
3) Open serial port to send data, then the data can be submitted onto our web page server.
3.2. Socket B Communication

USC520/USC540 support double socket communication mode, socket A and socket B. One serial port corresponding to two socket communication mode can be realized through setting the parameter of socket B. But socket B is only used for transparent transmission and only worked as TCP client or UDP client.

Set USC520/USC540 as double socket communication mode, the data of serial port will be transparently transmitted to socket A and socket B at one time. When data comes from socket A and socket B simultaneously, USC520/USC540 will transmit the data of socket A to serial port firstly, and then transmit the data of socket B once the data of socket A finished.

Communication example:

1. Set the parameter of socket A and socket B by web page
2. Set up socket A and socket B by “USR-TCP232-Test.exe”, and open serial port which connects to server.
3. Click “Send” on the software, data will be transmitted from socket A and socket B to serial port.
3.3. Modbus Gateway

3.3.1. Transmit modbus protocol in transparent mode

USC520/USC540 supports the transmission of modbus protocol in transparent mode

3.3.2. Modbus RTU to Modbus TCP

USC520/USC540 support Modbus RTU to Modbus TCP, settings as below:

2. Select “ModbusTCP” on the red color.
3. Click to save the parameter.
4. Check and verify Modbus RTU to Modbus TCP through modbus Poll and Modbus Slave

5. Setting modbus software is as below:

6. Click OK once configuration finished, update the data of modbus slave and modbus data will also be updated.
3.3.3. Modbus Active Query Function

Modbus active query function can be realized through the serial heartbeat packet function of USC520/USC540.

1. Open serial heartbeat packet function via web page, query command is heartbeat packet data, example:

![Heartbeat Packet Configuration](image)

2. Set modbus slave software, refer to the following:
3. Return result for query command is as drawing:
3.3.4. Modbus Polling Function

USC520/USC540 support modbus polling function under TCP server mode. Set USC520/USC540 as Modbus Poll, support multiple host polling to check parameter.

1. Multiple host polling is realized through virtual com.
   ① Set parameter of USC520/USC540 via web page, make sure to select ModbusTCP Poll and set overtime

   ![Parameter Setting](image)

   ② Open virtual com software to setup serval ports and connects to USC520/USC540
③ Open modbus slave software, choose the serial port which connected with USC520/USC540. Run modbus Poll software and make port-forwarding with this serial port.

④ The value of modbus poll will also be changed when you revise the value of modbus slave.

2. Conduct Modbus Polling through Modbus TCP to Modbus RTU

① Set USC520/USC540 via software, select Modbus TCP
② Open modbus slave software, choose the serial port which connected with USC520/USC540. Run modbus polling software to select network mode and connect with USC520/USC540.

③ The value of modbus poll will also be changed when you revise the value of modbus slave.

3. Modbus Polling supports 8 host query for the most, more in the near future. Need set polling time properly
when using modbus polling function. If polling interval is too short and baud rate is too low for the process of polling command which might lead to conflict between command circle and polling time.

3.4. Extra functions

3.4.1. DHCP

DHCP is obtaining IP address automatically. USC520/USC540 IP obtaining have 2 types: DHCP and static IP. It is static IP 192.168.0.7 by default.

DHCP is effective after change to DHCP and restart. When USC520/USC540 connects to router or device assigning IP, it require IP address from host within network, which takes about 5-15 seconds. Then you can search N520’s IP address. It is convenient for setting different IP address in different environment.

<Note> Don’t set DHCP when USC520/USC540 connected to PC directly because generally PC don’t have the ability of assigning IP. Otherwise, USC520/USC540 cannot transmit data normally, but wait for IP.

3.4.2. DNS

USC520/USC540 access the domain name or dynamic domain name when work under Client mode. The length of domain name must be less than 30 bytes. USC520/USC540 will analysis the domain name constantly if cannot connect to destination server.

When server’s IP address is dynamics, DNS make USC520/USC540 ’ parameter no changes if according IP doesn't change no matter how server IP address changes.
3.4.3. Self-Defined Heartbeat Function

USC520/USC540 supports heartbeat function. Heartbeat is divided into network heartbeat and serial port heartbeat.

① Set Network heartbeat through web page

Drawing 26 - Self-Defined Heartbeat Function
3.4.4. Self-Defined Registration Package Function

USC520/USC540 supports self-defined registration package function and also supports to send self-defined registration package after connection establishment, meanwhile, it supports to send registration package when sending data. One example as blow:

① Configure the relevant parameters via web page
② Test result:
3.4.5.  Web to serial

Web to serial function can make interaction between webpage and serial.

1.  Set port 6432 as default.
2.  Open webpage and click “web to serial”. It pops up “connect success” then can send/receive data. Open USR-TCP232-TEST Software, configure serial parameter and click “Open”.
3.  Click “send ASCII data”, COM can receive data. Click “Send” in TEST Software, webpage can receive data.

Web to serial needs user’s webpage programming ability. Design webpage, request own device’s data and process data then reveal the results on webpage. According to chapter 4.1.17 Customized Webpage, can download revised webpage into USC520/USC540.

1.  Build a connection and connect to USC520/USC540

   ```javascript
   function connectx(){
     try{
       socket=new WebSocket('ws://'+window.location.host+':6432');
       socket.binaryType = "arraybuffer";
     }catch(e){
       alert('error');
       return;
     }
     socket.onopen = sOpen;
   }
   ```
function sMessage(msg)

3. Send data function
   function send()

3.4.6. Network Printing Function

Network printing function is similar to printer server, it can be realized by the previous serial printer through the existed printing driver.

Testing Method:
① Configure the parameter, set work mode as “TCP Server”, local port number “9100”, and have to choose “Net Buffer” and “PRINT”. Others do not need to be chosen.
② Set Printer Driver

Click next and input the USC520/USC540'IP address, then keep clicking next til finished

③ Serial port connects to the printer, open a word file to print
3.4.7. Serial Port Packaging Mechanism

USC520/USC540 can configure serial port packaging time and serial port packaging length. USC520/USC540 will make packaging for the data of serial port according to the packaging length and packaging time in the transparent transmission mode.

Example for judgment of packaging time and packaging length:

① Set packaging time as 10ms, packaging length as 512 bytes

When serial port received data, USC520/USC540 will package and send it to network if the interval time of receiving data is over than 10ms or data length is more than 512.

② If the value of packaging time or packaging length is 0, the packaging rule is effective for non-zero one.

③ Set packaging time and length as 0. USC520/USC540 will conduct default packaging time when packaging time is set as 0ms. Namely, when serial port receiving data, USC520/USC540 will package and send the data to network if interval time more than packaging time of sending 4 bytes. For example, baud rate 115200, packaging time for 4 bytes is \( T = 0.4 \text{ms} \), when the calculated value is smaller than 0.1ms, packaging time can be calculated as 0.1ms.

\[ T = \frac{1}{\text{baud rate} \times 10^4} \]

3.4.8. Flow Calculation

When USC520/USC540 receives data from network and then send to serial port, as the limit of serial port
speed, user have to control the flow, if not the problem of data overflow on serial port side will occur. So data flow is required to calculated when sending data from network to serial port.

Example:

Network sends data in m bytes every n seconds. Method of checking if there is overflow: (Supposed network condition is good and network data transmission time is negligible)

1. If there is no overflow, m bytes data must be transmitted within n seconds, then the transmitting time of M bytes data:

\[ T = \frac{1}{\text{Baud Rate}} \times 10 \times m \]

If n > 2T, then data will not overflow, USC520/USC540 can work normally. Just need keep n>T under baud rate 9600.

3.4.9. Synchronous baud rate (RFC2217)

For encryption during data transmission, devices change data bytes, baud rate, parity and so on. USC520/USC540 supports revise serial parameter accordingly.

Synchronous baud rate is named RFC2217. USR Similar RFC2217 make adjustments on the basis of RFC2217 protocol to improve accuracy of transmission.

Protocol length is 8 bytes. And values taken for example is in HEX:

<table>
<thead>
<tr>
<th>Name</th>
<th>Packet Header</th>
<th>Baud Rate</th>
<th>Bytes parameter</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Explanation</td>
<td>reduce misjudgment</td>
<td>High is in front, smallest is 600(00 02 58)</td>
<td>data bytes, baud rate, parity</td>
<td>Remove 4 bits of header and ignore the high bit</td>
</tr>
<tr>
<td>(115200, N,8,1)</td>
<td>55 AA 55</td>
<td>01 C2 00</td>
<td>03</td>
<td>46</td>
</tr>
<tr>
<td>(9600, N,8,1)</td>
<td>55 AA 55</td>
<td>00 25 80</td>
<td>03</td>
<td>28</td>
</tr>
</tbody>
</table>

Serial parameter bit:

<table>
<thead>
<tr>
<th>Bit #</th>
<th>Explanation</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0</td>
<td>Data bit selection</td>
<td>00</td>
<td>5 bits</td>
</tr>
<tr>
<td>01</td>
<td>6 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>1 bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>Disable Parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>Enable Parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>ODD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>EVEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Mark</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using methods:
1. USR-TCP232-M4,E45 Setup software, click “Synchronous baud rate (RFC2217)”.
2. When serial parameter changes is needed, it send RFC 2217 packet. USC520/USC540 receive the command from network and revise serial parameter accordingly.

3.4.10. KeepAlive

When USC520/USC540’s network is abnormal, it can judge the status in time and disconnect. And connect to server once network recovers.

3.4.11. Device ID

The function have 2 types: send ID once connection and send ID once sending data. It is used to condition that need register packet or need packet header/tail for normal transmission.

3.4.12. Webpage Port

USC520/USC540 has built-in webpage server and the port is 80. Also the port can be revised and visit the web via revised port.

3.4.13. Revise MAC

User can check software’s MAC address. USC520/USC540 MAC is Globally Unique. Also it support
customized MAC.

3.4.14. Firmware Upgrade

Firmware upgrade is fulfilled via network. For details, please refer to Chapter 5.1 configure parameter with configuration software.

3.4.15. Hardware Flow Control RTS/CTS & XON/XoFF

RTS/CTS: USC520/USC540 supports serial RTS/CTS Hardware flow control function. It is disabled by default. Don't enable it if device doesn't support Hardware flow control. This function is only run under RS232 port.

XoN/XoFF: USC520/USC540 support software flow control function. It is disabled by default. It can be selected under RS232 port or RS422 port.

3.4.16. Reset

Shut off power first.
Press the reset button and supply power. Then keep pressing the reset button for 5 seconds.

4. Setting Protocol

USC520/USC540 includes two protocols: network setting protocol and serial port setting protocol.
4.1. Network Setting Protocol

4.1.1. Set Parameter Process

1. Build SOCKET:
   Build UDP SOCKET, destination IP: 55.255.255.255, destination port: 1901. Low is in front.
2. Setting command process:
   ① The network send searching command
   ② USC520/USC540 returns IP address and MAC
   ③ The network read USC520/USC540’s parameter
   ④ Organize setting command according to MAC, known user name/password and parameter to be configured.
   ⑤ Send setting command
   ⑥ USC520/USC540 returns “correct setting”
   ⑦ Host PC send “save setting” command
   ⑧ USC520/USC540 returns “correct”
   ⑨ Restart command
   ⑩ USE-N520 returns “correct setting”

4.1.2. Setting Command Content

Command Look-up List:

<table>
<thead>
<tr>
<th>Function</th>
<th>Header</th>
<th>Length</th>
<th>command</th>
<th>MAC (6 bytes)</th>
<th>User name/password (12bytes)</th>
<th>Parameter</th>
<th>Parity (sum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>search</td>
<td>FF</td>
<td>01</td>
<td>01</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>02</td>
</tr>
<tr>
<td>reset</td>
<td>FF</td>
<td>xx</td>
<td>02</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>-</td>
<td>xx</td>
</tr>
<tr>
<td>read</td>
<td>FF</td>
<td>xx</td>
<td>03</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>-</td>
<td>xx</td>
</tr>
<tr>
<td>settings</td>
<td>FF</td>
<td>xx</td>
<td>04</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>-</td>
<td>xx</td>
</tr>
<tr>
<td>Save</td>
<td>FF</td>
<td>xx</td>
<td>05</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>Basic parameter</td>
<td>xx</td>
</tr>
<tr>
<td>settings</td>
<td>FF</td>
<td>xx</td>
<td>06</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>COM parameter</td>
<td>xx</td>
</tr>
<tr>
<td>Com 0</td>
<td>FF</td>
<td>xx</td>
<td>07</td>
<td>[MAC]</td>
<td>[username] [password]</td>
<td>COM parameter</td>
<td>xx</td>
</tr>
<tr>
<td>settings</td>
<td>FF</td>
<td>xx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Notice: Check bit is sum check, starts from length byte (including length) to adding before checking (not including checking), result is check value, only low byte is remained.

1. Command examples
   ① Search command example
   Search command is set to:
   FF 01 01 02
   Sum check:  02 = 01 + 01
   ② Reset command example
   FF 13 02 d8 b0 04 c9 61 64 6d 69 6e 00 61 64 6d 69 6e 00 c8
   Sum check:
   C8 = 13 + 02 + ... + 6E + 00
   User name and password both are 5 bytes+00 bits 0 for the lack.
   ③ Read settings command example
   Send (16 bytes): FF 13 03 AC CF 23 66 66 67 61 64 6D 69 6E 00 61 64 6D 69 6E 00 F9
   ④ Save reading settings command example
   Send (16 bytes): FF 13 04 AC CF 23 66 66 67 61 64 6D 69 6E 00 61 64 6D 69 6E 00 FA

2. Some commands detailed annotation
   ① Basic setting parameter command
### Basic Parameter:

<table>
<thead>
<tr>
<th>Name</th>
<th>Byte</th>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ucSequenceNumber</td>
<td>1</td>
<td>xx</td>
<td>Write the read values</td>
</tr>
<tr>
<td>ucCRC</td>
<td>1</td>
<td>xx</td>
<td>Write the read values</td>
</tr>
<tr>
<td>ucVersion</td>
<td>1</td>
<td>xx</td>
<td>Write the read values</td>
</tr>
<tr>
<td>ucFlags</td>
<td>1</td>
<td>80</td>
<td>IP address type: 0 in 8th bit: DHCP; 1 in 8th bit: Static IP</td>
</tr>
<tr>
<td>usLocationURLPort</td>
<td>2</td>
<td>20 19</td>
<td>Write the read values</td>
</tr>
<tr>
<td>usHTTPServerPort</td>
<td>2</td>
<td>50 00</td>
<td>HTTP server port</td>
</tr>
<tr>
<td>ucUserFlag</td>
<td>1</td>
<td></td>
<td>Write the read values</td>
</tr>
<tr>
<td>ulStaticIP</td>
<td>4</td>
<td>38 00 A8 C0</td>
<td>Static IP</td>
</tr>
<tr>
<td>ulGatewayIP</td>
<td>4</td>
<td>01 00 A8 C0</td>
<td>Gateway</td>
</tr>
<tr>
<td>ulSubnetMask</td>
<td>4</td>
<td>00 FF FF FF</td>
<td>Subnet Mask</td>
</tr>
<tr>
<td>ucModName</td>
<td>16</td>
<td>55 53 52 2D 54 43 50 32 33 32 2D 45 00 00</td>
<td>USC520/USC540 name</td>
</tr>
<tr>
<td>username</td>
<td>6</td>
<td>61 64 6D 69 6E 00</td>
<td>username</td>
</tr>
<tr>
<td>password</td>
<td>6</td>
<td>61 64 6D 69 6E 00</td>
<td>password</td>
</tr>
<tr>
<td>ucNetSendTime</td>
<td>1</td>
<td></td>
<td>Write the read values</td>
</tr>
<tr>
<td>uiId</td>
<td>2</td>
<td>01 00</td>
<td>Device ID</td>
</tr>
<tr>
<td>ucIdType</td>
<td>1</td>
<td>0</td>
<td>Device ID type (0~3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: no use</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: send id when connect</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: send id when send data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: both</td>
<td></td>
</tr>
<tr>
<td>ucUserMAC</td>
<td>6</td>
<td>FF FF FF FF FF FF</td>
<td>MAC</td>
</tr>
<tr>
<td>ucReserved</td>
<td>8</td>
<td></td>
<td>Unused</td>
</tr>
</tbody>
</table>

Example:

```
FF 56 05 AC CF 23 66 66 67 61 64 6D 69 6E 00 61 64 6D 69 6E 00 61 66 03 80 20 19 50 00 02 07 00 A8 C0 01 00 A8 C0 00 FF FF FF 55 53 52 2D 54 43 50 32 33 32 2D 45 34 35 00 00 61 64 6D 69 6E 00 61 64 6D 69 6E 00 02 01 00 00 AC CF 23 66 66 67 00 48 54 54 50 2F 31 2E 1C
```
### Port settings parameter command

#### Port parameter:

<table>
<thead>
<tr>
<th>Name</th>
<th>bytes</th>
<th>example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ulBaudRate</td>
<td>4</td>
<td>00 C2 01 00</td>
<td>Baud Rate</td>
</tr>
<tr>
<td>ucDataSize</td>
<td>1</td>
<td>08</td>
<td>COM data bits (0x05/0x06/0x07/0x08)</td>
</tr>
<tr>
<td>ucParity</td>
<td>1</td>
<td>01</td>
<td>COM parity 1: no, 2: odd, 3: even, 4: mark, 5: space</td>
</tr>
<tr>
<td>ucStopBits</td>
<td>1</td>
<td>01</td>
<td>COM stop bit (0x01/0x02)</td>
</tr>
<tr>
<td>ucFlowControl</td>
<td>1</td>
<td>01</td>
<td>COM flow control (0x01; no, 0x03:HW)</td>
</tr>
<tr>
<td>ulTelnetTimeout</td>
<td>4</td>
<td>00 00 00 00</td>
<td>Network reconnection time</td>
</tr>
<tr>
<td>usTelnetLocalPort</td>
<td>2</td>
<td>17 00</td>
<td>Local Port</td>
</tr>
<tr>
<td>usTelnetRemotePort</td>
<td>2</td>
<td>17 00</td>
<td>Remote Port</td>
</tr>
<tr>
<td>uiTelnetURL</td>
<td>30</td>
<td>31 39 32 2E 31 36 38 2E 30 2E 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td>
<td>IP address send in ASCII. Example: 192.168.0.1</td>
</tr>
<tr>
<td>ulTelnetIPAddr</td>
<td>4</td>
<td>00 00 00 00</td>
<td>Not adopted</td>
</tr>
<tr>
<td>ucFlags</td>
<td>1</td>
<td>02</td>
<td>Enable MODBUS_TCP: 0x010(bit2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Enable 2217: 0x08(bit3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Enable USR cloud: 0x010(bit4)</td>
</tr>
<tr>
<td>ucWorkMode</td>
<td>1</td>
<td>03</td>
<td>Working mode: 0: UDP, 1: TCP Client, 2: UDP Server, 3: TCP Server, 4: HTTPD Client</td>
</tr>
<tr>
<td>uiPackLen</td>
<td>4</td>
<td>C8 00 00 00</td>
<td>COM pack length</td>
</tr>
<tr>
<td>ucPackTime</td>
<td>1</td>
<td>0A</td>
<td>COM pack time</td>
</tr>
<tr>
<td>ucTimeCount</td>
<td>1</td>
<td>91</td>
<td>Write the read values</td>
</tr>
<tr>
<td>TCP server type</td>
<td>1</td>
<td>1</td>
<td>Write the read values</td>
</tr>
<tr>
<td>ucReserved</td>
<td>4</td>
<td>Casual value saved</td>
<td></td>
</tr>
</tbody>
</table>

Example:

```
FF 52 06 AC CF 23 66 66 67 61 64 6D 69 6E 00 61 64 6D 69 6E 00 00 C2 01 00 08 01 01 01 00 00 00 00 17 00 17 00 31 39 32 2E 31 36 38 2E 30 2E 32 30 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 03 C8 00 00 00 0A 95 11 00 00 00 42
```
4.1.3. Commands' Return Content

1. Return results of search command

Return command:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Name</th>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TAG_STATUS</td>
<td>FF</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Packet_length</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CMD_DISCOVER_TARG</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Board_type</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Board_ID</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>5~8</td>
<td>Client_IP_address</td>
<td>C0 A8 00 07</td>
<td>Device IP(High in front)</td>
</tr>
<tr>
<td>9~14</td>
<td>MAC_address</td>
<td>AC CF 23 20 FE 3D</td>
<td>Device MAC(High in front)</td>
</tr>
<tr>
<td>15~18</td>
<td>Firemware_version</td>
<td>D0 07 12 34</td>
<td>D0 07: device version# (low in front) 12 34: encrypted version</td>
</tr>
<tr>
<td>19~34</td>
<td>Application_title</td>
<td>55 53 52 2D 54 43 50 32 33 32 2D 35 30 30 00 00</td>
<td>Device name</td>
</tr>
<tr>
<td>35</td>
<td>checksum</td>
<td>F0</td>
<td>checksum</td>
</tr>
</tbody>
</table>

Example:
Return results of search command(36 bytes)
FF 24 01 00 4B C0 A8 00 4D D8 B0 4C 00 04 C9 DD 07 01 00 55 53 52 2D 54 43 50 32 33 32 2D 34 30 31 00 00 EF
The method of the check is as follow:
0xEF = 00 - FF - 24 - 01 - 00 - 4B - ... - 31 - 00 - 00

2. Return results of reset command
Response(4 bytes): FF 01 02 4B, if user name and password are right, 4B = 'K'
FF 01 02 45, if user name and password are wrong, 45 = 'E'

3. Return results of read command
Description:
Return all parameter of USC520/USC540 network. 193 bytes in total, no parity, no protocol, return
parameter directly.

Returned content: 193 (basic parameter+serial parameter+serial parameter)

4. Return results of save settings command
   If settings are correct, it returns:
   FF 01 04 4B

5. Return results of basic settings command
   FF 01 05 4B

6. Others return results
   Sum check fault returns 'E' + right parity
   Correct execution: FF 01 CMD 'K'
   User name/password fault returns: FF 01 CMD 'P'
   Others faults return: FF 01 CMD 'E'

4.2. Serial Setting Protocol

4.2.1. AT Command

AT Command List:

<table>
<thead>
<tr>
<th>Command</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(blank)</td>
<td>Blank Command</td>
</tr>
<tr>
<td>E</td>
<td>Open/close echo function</td>
</tr>
<tr>
<td>Z</td>
<td>Re-start device</td>
</tr>
<tr>
<td>VER</td>
<td>Query version number</td>
</tr>
<tr>
<td>ENTM</td>
<td>Enter to transparent transmission mode</td>
</tr>
<tr>
<td>MAC</td>
<td>Query/set MAC</td>
</tr>
<tr>
<td>UART</td>
<td>Set/query serial port parameter</td>
</tr>
<tr>
<td>SOCKMN</td>
<td>Set SOCK parameter</td>
</tr>
<tr>
<td>DHCPEN</td>
<td>Open/close automatic obtaining IP address</td>
</tr>
<tr>
<td>WANN</td>
<td>Open/close WAN port parameter</td>
</tr>
<tr>
<td>WEBU</td>
<td>Query/set web page user’s name and password</td>
</tr>
<tr>
<td>WEBPOR</td>
<td>Query/set web page port number</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Query/set searching key words</td>
</tr>
</tbody>
</table>

Table 1 AT Command List
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAR</td>
<td>Factory reset</td>
</tr>
<tr>
<td>UART</td>
<td>Set/query serial port 0 parameter</td>
</tr>
<tr>
<td>UART1</td>
<td>Set/query serial port 1 parameter</td>
</tr>
<tr>
<td>SOCK</td>
<td>Set/query port 0 network SOCKETA parameter</td>
</tr>
<tr>
<td>SOCK1</td>
<td>Set/query port 1 network SOCKETA parameter</td>
</tr>
<tr>
<td>SOCKB</td>
<td>Set/query port 0 network SOCKETB parameter</td>
</tr>
<tr>
<td>SOCKB1</td>
<td>Set/query port 1 network SOCKETB parameter</td>
</tr>
<tr>
<td>MAC</td>
<td>Query MAC parameter</td>
</tr>
<tr>
<td>WRMAC</td>
<td>Set MAC parameter</td>
</tr>
<tr>
<td>DNS</td>
<td>Set/query domain name analyzing address</td>
</tr>
<tr>
<td>WANN</td>
<td>Set/query module IP address, gateway, subnet mask parameter</td>
</tr>
</tbody>
</table>
4.2.2. Enter AT Command Mode

① Send +++ to USC520/USC540 through serial port
② USC520/USC540 return ‘a’
③ Need reply “a” within 3s once received previous ‘a’
④ Return+ok to enter AT command mode

4.2.3. AT Command details

1. AT+E
   - Function: Set/query echo setting
   - Format:
     - Query
       AT+E <CR>
       <CR><LF>+OK=<on/off><CR><LF>
     - Set
       AT+E=<on/off><CR>
       <CR><LF>+OK<CR><LF>
   - Parameter:
     ① on: Open echo function, and input AT command (Echo on)
     ② off: Close echo function. (Echo off)

2. AT+ENTM
   - Function: Enter transparent transmission mode (Exit command mode)
   - Format:
     - Set
       AT+ENTM<CR>
       <CR><LF>+OK<CR><LF>
   - Parameter: N/M
     Change work mode from command mode to transparent transmission mode after run this command correctly

3. AT+Z
   - Function: Re-start USC520/USC540
   - Format:
     - Set
       AT+Z<CR>
       <CR><LF>+OK<CR><LF>
   - Parameter: N/M
     <Notice>: USC520/USC540 will re-start after run this command correctly.
4. **AT+VER**
   - **Function:** Set/query firmware version
   - **Format:**
     - **Query**
       ```
       AT+VER<CR>
       <CR><LF>+OK=<ver><CR><LF>
       ```
     - **Parameter:**
       - **ver:** Set/Query the firmware version of USC520/USC540
       - General version: AA.BB.CC: AA stands for big version, BB stands for small version number, CC stands for hardware version C.C
       - Custom version: AA.BB.CC.DD-ID; DD stands for customer’s version, ID stands for customer’s ID number

5. **AT+MAC**
   - **Function:** Query MAC
   - **Format:**
     - **Query**
       ```
       AT+MAC<CR>
       <CR><LF>+OK=<mac><CR><LF>
       ```
     - **Parameter:**
       - **mac:** MAC Address of USC520/USC540 (Example 01020304050A);

6. **AT+WRMAC**
   - **Function:** Set MAC
   - **Format:**
     - **Set**
       ```
       AT+WRMAC=<mac><CR>
       <CR><LF>+OK<CR><LF>
       ```
     - **Parameter:**
       - **mac:** MAC Address of USC520/USC540 (Example 01020304050A);

7. **AT+WEBU**
   - **Function:** Set/query user’s name and password of web page
   - **Format:**
     - **Query**
       ```
       AT+WEBU<CR>
       <CR><LF>+OK=<username,password><CR><LF>
       ```
     - **Set**
       ```
       AT+WEBU<CR>
       <CR><LF>+OK=<username,password><CR><LF>
       ```
     - **Parameter:**
       - **username:** user’s name, supports max 5 bytes. Does not support blank
8. AT+WEBPORT
   - Function: Set/query web server port of USC520/USC540
   - Format:
     - Query
       \[ \text{AT+WEBPORT<CR>} \]
       \[ <CR><LF>+OK=<port><CR><LF> \]
     - Set
       \[ \text{AT+WEBPORT<CR>} \]
       \[ <CR><LF>+OK =<port><CR><LF> \]
   - Parameter:
     - port: web server’s port of USC520/USC540, default 80;

9. AT+SEARCH
   - Function: Set/query search port and key words of USC520/USC540 inside LAN
   - Format:
     - Query
       \[ \text{AT+SEARCH<CR>} \]
       \[ <CR><LF>+OK=<port,keywords><CR><LF> \]
     - Set
       \[ \text{AT+SEARCH<CR>} \]
       \[ <CR><LF>+OK =<port,keywords><CR><LF> \]
   - Parameter:
     - port: serching port of USC520/USC540, default 48899
     - keywords: searching key words of USC520/USC540, Default: www.usr.cn (Max 20 bytes).

9. AT+CLEAR
   - Function: Restore factory setting
   - Format:
     - Set
       \[ \text{AT+CLEAR<CR>} \]
       \[ <CR><LF>+OK<CR><LF> \]
   - Parameter: N/M

10. AT+UART
    - Function: Set/query the parameter of UART0
    - Format:
      - Query
        \[ \text{AT+UART<CR>} \]
        \[ <CR><LF>+OK=<baudrate,data_bits,stop_bit,parity,flowctrl, uartnum><CR><LF> \]
      - Set
        \[ \text{AT+UART=<baudrate,data_bits,stop_bit,parity,flowctrl, uartnum><CR><LF>} \]
        \[ <CR><LF>+OK<CR><LF> \]
Parameter:
- baudrate: baud rate
  - 9600, 19200, 38400, 57600, 115200, 230400, 380400, 460800
- data_bits: data bits 5, 6, 7, 8
- stop_bits: stop bits 1, 1.5, 2 (which module supports)
- parity: check bit
  - NONE (no check bit)
  - EVEN (even check)
  - ODD (odd check)
  - MASK (1 check)
  - SPACE (0 check)
- flowctrl: hardware flow control (CTS RTS)
  - NFC: no hardware flow control
  - FC: hardware flow control
  - 485: supports 485, after opening, RS_EN pin and RTS pin are the same
- uartnum: 0

11. AT+UART1
   - Function: Set/query UART0 interface parameter

12. AT+SOCKMN
   (M: socket number, A~H: N: corresponding serial port number, 1~8. Omit MN numbers for single serial port and single socket)
   - Function: Set/query network protocol parameter format:
     - Query
       ```
       AT+SOCKMN<CR>
       <CR><LF>+OK=<protocol,IP,port><CR><LF>
       ```
     - Set
       ```
       AT+SOCKMN=< protocol,IP,port ><CR>
       <CR><LF>+OK<CR><LF>
       ```
   - Parameter:
     - Protocol: protocol type
       - TCPS: TCP server
       - TCPC: TCP client
       - UDPS: UDP server
       - UDPC: UDP client
       - TCPA: TCP auto
       - WEBUART: websocket, transmission from webpage to serial port
     - IP: Server's IP address when USC520/USC540 works as "CLIENT"
     - Port: Protocol port, decimal number, smaller than 65535

13. AT+DHCPEN
   - Function: Open/close DHCP Server;
Format:

Query

AT+DHCPEN<CR>
<CR><LF>+OK=<Type><CR><LF>

Set

AT+DHCPGW=<Type><CR>
<CR><LF>+OK<CR><LF>

Parameter

Type:

On: open DHCP server function
Off: close DHCP server function (USC520/USC540 works as wireless switch at this time)

14. AT+WANN

Function: Set/query obtained WAN port IP (DHCP/STATIC);

Format:

Query

AT+WANN<CR>
<CR><LF>+OK=<mode,address,mask,gateway><CR><LF>

Set

AT+WANN=<mode,address,mask,gateway><CR>
<CR><LF>+OK<CR><LF>

Parameter:

mode: network IP mode.
static: static IP
DHCP: dynamic IP (address, mask, gateway parameter is omitted)

address: IP address.
mask: subnet mask.
gateway: gateway address.

5. Parameter Configuration

It is setup software configuration, webpage configuration and serial configuration.
How to configure:
Revise user name/password → set IP access method → serial parameter → USC520/USC540 work mode → work mode related parameter

5.1. Software Configuration

To make sure setup software normal running, please check the below firstly:
1. USC520/USC540 and setup software PC are within same LAN.
2. Disable the anti-virus software and firewall on PC.
3. Disable network card nothing to do with this testing.

Download the [USR-TCP232-M4&E45] Setup software.
Search device and all USC520/USC540 devices within LAN. It includes IP, name, MAC and version.

Software Configuration—Search
1. Click ‘Device’ on the top of the program and then check user name/password via ‘User config’. If it is correct, it reveals USC520/USC540 information. If not, it pops up retype window, click “Confirm”. User name and password is admin by default.
2. Basic parameter configuration

Click “Full Show” and all basic parameter is revealed.

Set the parameter as needs and click “Base Save” then can set successfully.
Software Configuration -- Full Show
Software Configuration-Base Save

- Websocket port: refer to Chapter 4.1.8.3 Webpage to serial
- Webpage port: it is 80 by default when visit webpage.
- Device ID: refer to Chapter 4.1.8.6 Device ID
- Device ID type: sending ID type
- User name: Authentication Code for revising parameter to avoid other users within same LAN revising it.
- Password: same as user name.
- Device Name: USC520/USC540’s name an be revised.
- MAC address: USC520/USC540’s MAC
- IP address type: Static and DHCP
- USC520/USC540 static IP: same segment with router.
- Subnet Mask: 255.255.255.0 by default.
- Gateway: it is router IP generally, can transmit cross network segment and DNS if set correctly.
3. Port configuration (Port1 / Port2 configuration)
Click the COM to set, revise parameter then click “Save COM1”.

**Software Configuration-COM 1 Configuration**

- **Serial Baud rate**: it can be standard or customized.
- **Parity/Data/Stop**: serial parameter.
- **Serial Flow control**: None/Hardware/XON XOFF, None for no flow control, Hardware for hardware flow control, XON/XOFF for software hardware flow.
- **Work Mode**: TCP Server /TCP Client/HTTPD Client/UDP Client/UDP Server
- **Destination IP/Port**: IP connected when USC520/USC540 works as client (TCP Client/HTTPD Client/UDP Client)
- **Local Port**: port USC520/USC540 to connect. Advice to set it to “0” when USC520/USC540 works under TCP Client for connection with Random port.
- **TCP Server Type**: No.
- **Modbus TCP**: set this when Modbus TCP to Modbus RTU is needed.
- **Serial pack time**: relate to serial unpacking mechanism.
Serial pack length: relate to serial unpacking mechanism.

Similar RFC2217: Please refer to Chapter 3.5.10 Similar RFC2217

4. Firmware Upgrade

If USC520/USC540 need to upgrade with new firmware, please contact USR sales. During firmware upgrade, USC520/USC540 connects to PC directly. PC Upgrade via Wi-Fi is prohibited.
Firmware Upgrade