USR-TCP232-E45/M4 Cofig Protocol

Protocol

All data must be sent via UDP boardcast, local port 1901, sent to remote port 1901, broadcast address 255.255.255.255.

First PC send a command, the device reply.

1. Send Command

All the search or config command format list below.

Function	Head	Length (Command~ Parameter)	Command	MAC (6 byte)	User/Password (12 bytes)	Parameter	Check Byte (sum)
Search	FF	01	01				02
Reset	FF	xx	02	[MAC]	[usrname] [password]		xx
Read	FF	xx	03	[MAC]	[usrname] [password]		xx
Store	FF	xx	04	[MAC]	[usrname] [password]		xx
Basic Setting	FF	xx	05	[MAC]	[usrname] [password]	Basic param	xx
COM 0 setting	FF	xx	06	[MAC]	[usrname] [password]	Port0 param	xx
COM 1 setting	FF	xx	07	[MAC]	[usrname] [password]	Port1 param	xx
COM 2 setting	FF	xx	08	[MAC]	[usrname] [password]	Port2 param	xx
Expand setting							
Read temporary config	*FF*	*xx*	*0A*	*[MAC]*	*[usrname]* *[password]*		*xx*
MAC setting	*FF*	*xx*	*FE*	*[MAC]*			*xx*

1.1. About Checksum

Last byte is checksum, it calculated from length byte(length byte included), until the checksum byte(not include checksum itself), the result byte is checksum, checksum reserves only 1 byte.

1.2. Search command

Search command is fixed to such format

FF 01 01 02

The cheksum byte 02 is calculated as 02 = 01 + 01 .

1.3. Reset command

The command is used to reset a selected one device, one device only

Send:

FF 13 02 d8 b0 4c 00 04 c9 61 64 6d 69 6e 00 61 64 6d 69 6e 00 c8

Checksum C8 = 13 + 02 + ... + 6E + 00

Red part is module's mac address

Last 12 byte is the module's user name and password, if the string length less than 6 byte, rest byte must be filled with 0.

1.4. Read param

This command is used to read all parameters out of the selected device

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

Checksum F9 = 13 + 03 + AC + ... + 6E + 00

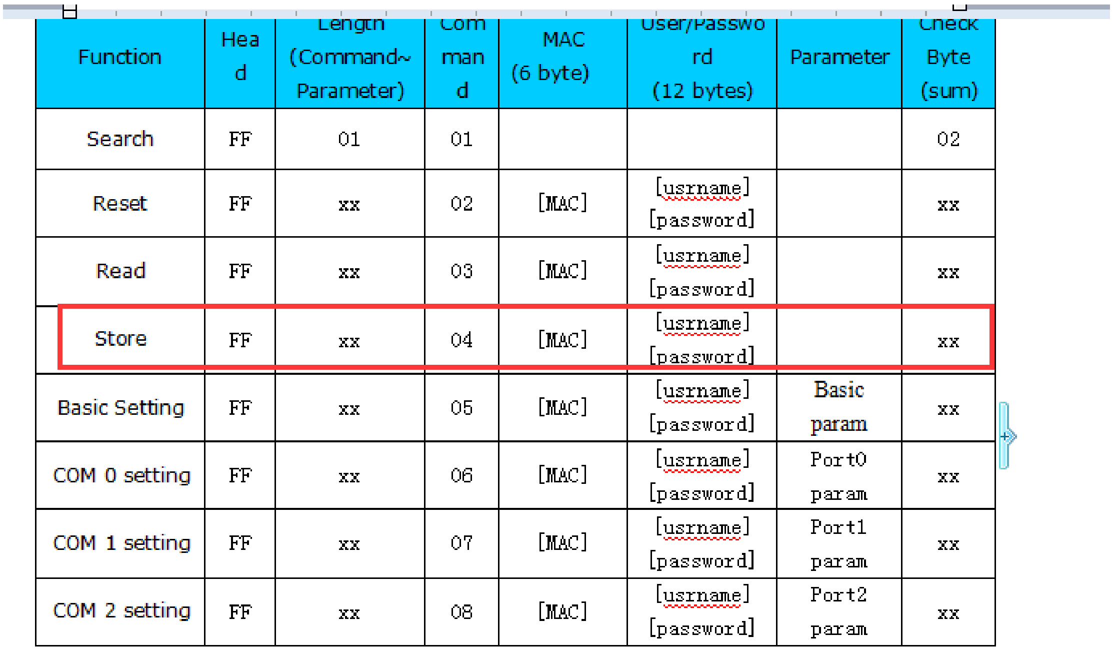
The green part is username, red part is password, if the string length less than 6 byte, rest byte must be filled with 0.

1.5. Store param

This command is used to save param into module.

Send:

FF 13 04 AC CF 23 66 66 67 61 64 6D 69 6E 00 61 64 6D 69 6E 00 FA



Function	Head	Length (Command~ Parameter)	Command	MAC (6 byte)	User/Password (12 bytes)	Parameter	Check Byte (sum)
Search	FF	01	01				02
Reset	FF	xx	02	[MAC]	[usrname] [password]		xx
Read	FF	xx	03	[MAC]	[usrname] [password]		xx
Store	FF	xx	04	[MAC]	[usrname] [password]		xx
Basic Setting	FF	xx	05	[MAC]	[usrname] [password]	Basic param	xx
COM 0 setting	FF	xx	06	[MAC]	[usrname] [password]	Port0 param	xx
COM 1 setting	FF	xx	07	[MAC]	[usrname] [password]	Port1 param	xx
COM 2 setting	FF	xx	08	[MAC]	[usrname] [password]	Port2 param	xx

1.6. Basic config

Basic config parameters, 67 byte total.

Name	Byte	Example	Instruction
ucSequenceNum	1		
ucCRC	1		
ucVersion	1		
ucFlags	1	80	IP address type: The eighth is 0: DHCP; 1: Statics IP
usLocationURLPort	2	20 19	UPNP port
usHTTPServerPort	2	50 00	HTTP service port
ucUserFlag	1		
ulStaticIP	4	38 00 A8 C0	Statics IP
ulGatewayIP	4	01 00 A8 C0	Gateway
ulSubnetMask	4	00 FF FF FF	Subnet mask
ucModName	16	55 53 52 2D 54 43 50 32 33 32 2D 45 00 00 00 00	Module name
username	6	61 64 6D 69 6E 00	Username

password	6	61 64 6D 69 6E 00	Password
ucNetSendTime	1		
uiId	2	01 00	Device ID
ucIdType	1	00	Device ID (0~3) 0:no use 1:send id when connect 2:send id when send data 3:both
ucUserMAC	6	FF FF FF FF FF FF	MAC address
ucReserved	8	Any will do	unused

Below is a sample command, send:

```
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
```

Checksum 1C = 56 + 05 + AC + ... + 2E.

Red part is module's mac address, followed with 12 byte username and password(rest byte filled 0)

The rest byte is basic parameters you want to config. Checksum is the last byte.

1.7. Port config

Each port's parameters is 63 byte, the is usually 3 port for each module. Port0, port1, port2.

Name	byte	sample	description
ulBaudRate	4	00 C2 01 00	baudrate
ucDataSize	1	08	databit(0X05/0x06/0x07/0x08)
ucParity	1	01	parity 1： no， 2： odd， 3： even， 4： mark， 5： space
ucStopBits	1	01	stopbit(0x01/0x02)
ucFlowControl	1	01	flowcontrol(0x01： no， 0x03： HW)
ulTelnetTimeout	4	00 00 00 00	Unused
usTelnetLocalPort	2	17 00	Local port
usTelnetRemotePort	2	17 00	Remote port
uiTelnetURL	30	31 39 32 2E 31 36 38 2E 30 2E 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ip address or domain name is sent with string format, such as "192.168.0.1", or "www.usr.cn".
ulTelnetIPAddr	4	00 00 00 00	unused
ucFlags	1	02 by default	unused
ucWorkMode	1	03	Work mode 0：UDP, 1：TCP Client, 2：UDP Server, 3：TCP Server, 4： HTTPD Client
uiPackLen	4	C8 00 00 00	Serial packet length
ucPackTime	1	0A	Serial packet time ms(10 by default)
ucTimeCount	1	91	
TCP server type	1	80 (by default)	Higher 4 bits: pls set this bit to 8 Lower 4 bits: tcp server type(this bit is only

			useful when module work as tcp server) 1: transparent transportation 2: send with ID(discard packet if there is no id) 3: send with ID(send to all client if there is no ID)
ucReserved	4(60)		Unused (usually 4 bytes , but 60 byte when u used a httpd client firmware)

Send:

```
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 08 03 C8 00 00 00 0A 95 11 00 00 00 00 42
```

Checksum: 42 = 52 + 06 + ... + 00

Red part is mac address, followed with username + password + portx param + checksum(1 byte)

1.8. program MAC

Attention, the mac address can only be programed once(when mac is default FFFFFFFFFFFF).

Send:

FF 07 FE 00 11 22 33 44 55 sum

Return(sucess)

FF 01 FE 4B

2. Return command

2.1. Search return

Byte	name	Example	Instruction
0	TAG_STATUS	FF	
1	Packet_length	24	
2	CMD_DISCOVER_TARGET	01	
3	Board_type	00	
4	Board_ID	00	
5~8	Client_IP_address	C0 A8 00 07	Device IP（High bit in front）
9~14	MAC_address	AC CF 23 20 FE 3D	Device MAC（High bit in front）
15~18	Firmware_version	D0 07 12 34	D0 07：Device version number （low bit in front） 12 34：Encrypted version ;The others is non encrypted version;Encrypted program upgrade directly in encryption version;Non encrypted version need to decrypt the encrypted program,then send
19~34	Application_title	55 53 52 2D 54 43 50 32 33 32 2D 35 30 30 00 00	Device name
35	checksum	F0	

2.2. Reset return

Return(4 byte)

FF 01 02 4B (if usename and password correct 4B = ’K’)

FF 01 02 45 (if username and pass correct ＝ ’E’)

2.3. Read return

Return all parameters of module, 256 byte in total, without checksum or header.

Must judge length of returned parameter.

If length is 193 byte(basic param + 2*port param), this means the module have 2 port.

If length is 256 byte(basic param + 3*port param), this means the module have 3 port.

returns(256 byte)：

```
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 31 20 00 00 05 01 01 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 00 00 00 00 00 00 00 C2 01 00 08 01 01 01 00 00 00 00 17 00 17 46 31 39 32
2E 31 00 00 00 00 05 01 01 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08
05 C8 00 00 00 0A 36 11 00 C2 01 00 08 01 01 01 00 00 00 00 1A 00 1A 00 31 39 32 74 31 36 38 2E
30 00 00 00 00 05 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 04 C8 00 00 00
0A 1A 11 00 C2 01 00 08 01 01 05 00 00 00 00 1D 00 1D 00 31 39 32 2E 31 36 38 74 30 2E 32 30 31
```

If your read command is incorrect, it will return,

```
FF 01 03 45
```

The return have no checksum

2.4. Store return

If successful, it will return

```
FF 01 04 4B
```

Basic config

If successful, it will return

```
FF 01 04 4B
```

Other return

Successfully execuated: FF 01 CMD 'K' (the CMD is command byte in your command)

Checksum error: return 'E' + checksum(the correct checksum)

Username or pass error: FF 01 CMD 'P'

Other error will return: FF 01 CMD 'E'

3. Sample command and return

3.1. Some sample command and return

Search

Send(4 byte search command)

```
Ff 01 01 02
Return (36 byte)
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
```

Reset module

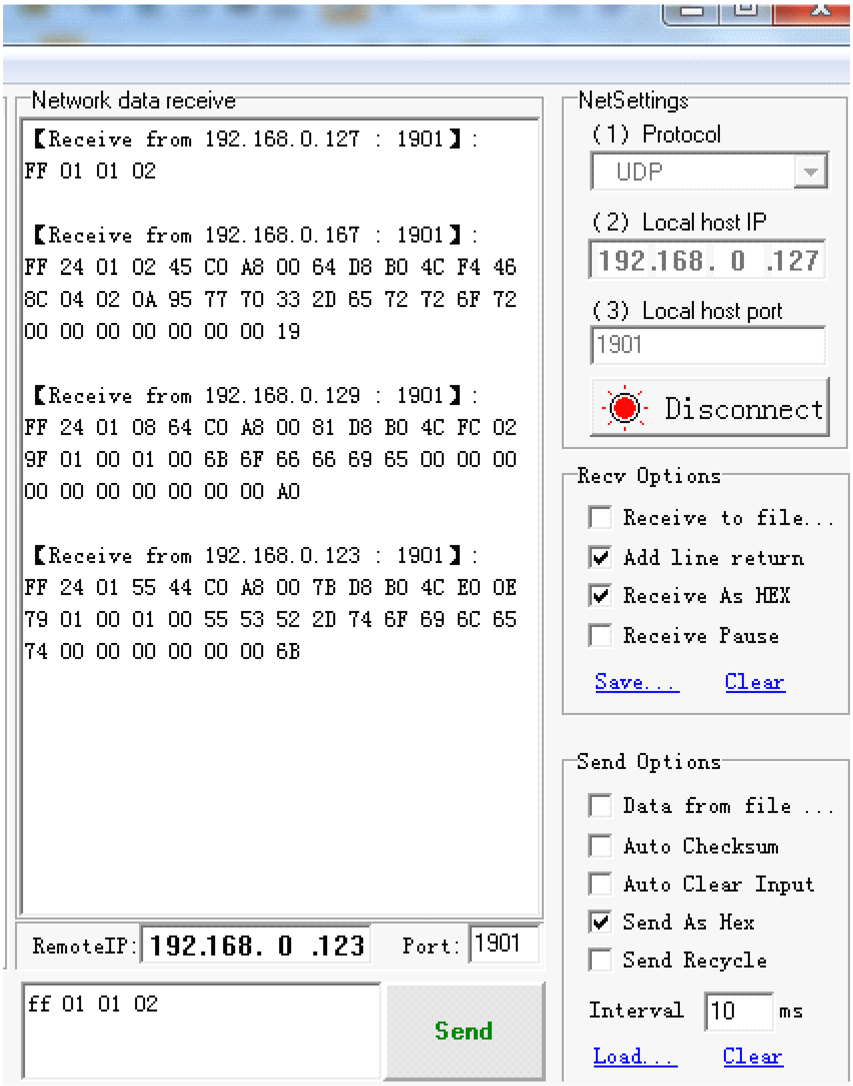
send

```
FF 13 02 d8 b0 4c 00 04 c9 61 64 6d 69 6e 00 61 64 6d 69 6e 00 c8
return(4 byte): FF 01 02 4B
```

3.2. Way to get message

If you want to get some command and return sample while you are config module via software, there is a way to get what you want.

Open a USR-TCP232-Test, and choose protocol UDP, listen on port 1901. When you are searching or config a module using another PC, you can get their interchange message in the receive window of Test software.



Network data receive

```
【Receive from 192.168.0.127 : 1901】:
FF 01 01 02

【Receive from 192.168.0.167 : 1901】:
FF 24 01 02 45 C0 A8 00 64 D8 B0 4C F4 46
8C 04 02 0A 95 77 70 33 2D 65 72 72 6F 72
00 00 00 00 00 00 00 19

【Receive from 192.168.0.129 : 1901】:
FF 24 01 08 64 C0 A8 00 81 D8 B0 4C FC 02
9F 01 00 01 00 6B 6F 66 66 69 65 00 00 00
00 00 00 00 00 00 00 A0

【Receive from 192.168.0.123 : 1901】:
FF 24 01 55 44 C0 A8 00 7B D8 B0 4C E0 0E
79 01 00 01 00 55 53 52 2D 74 6F 69 6C 65
74 00 00 00 00 00 00 6B
```

RemoteIP: 192.168. 0 .123 Port: 1901

ff 01 01 02

NetSettings

(1) Protocol: UDP

(2) Local host IP: 192.168. 0 .127

(3) Local host port: 1901

Recv Options

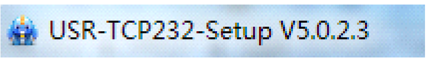
□ Receive to file...
☑ Add line return
☑ Receive As HEX
□ Receive Pause

Send Options

□ Data from file ...
□ Auto Checksum
□ Auto Clear Input
☑ Send As Hex
□ Send Recycle

Interval 10 ms

4. Serial config protocol

Get access to serial config mode, first, connect Reload(CFG) to GND.
Module config command format as following table, you can also use our setup software(serial config part is compatible to USR-TCP232-T24 setup  USR-TCP232-Setup V5.0.2.3) to generate and test config word, baud rate 9600, none parity, 8 data, 1 stop: 9600,n, 8, 1

No matter which baud is in previous, module will switch to 9600 in config mode, and send character U to com, to indicate module do in config mode. After receiving complete data package and check correctly, will reply K, if check incorrect, will reply E and module calculate parity bit, this bit is quite useful when test to send command manually. For other errors, for example incorrect package header or bit number, will reply only E.

Note: Need to release CFG to vacant or connect to VCC, to make module back to working mode (for E45/M4 series, release cfg (Reload) will make module saving parameters and restart)

Command package head

UART configuration command

55 BA - write Port 0 configuration, 55 BC -read Port 0 configuration

55 C6 - write basic configuration, 55 c6 - read basic configuration

When write port config, all data bit according to the following table; when read port config, send package header is enough. Example: send 55 BC, will read port 0 config parameters.

4.1. Param table 1

Data bit meaning as listed below:

Function	Bit	Instruction	Example	Hex, low in front
Packet head	2	55 BA	Packet head	55 BA
Destination IP	4	Connected target IP	192.168.0.201	C9 00 A8 C0
Destination port	2	Connected target Port	8234	2A 20
Module IP	4	Module IP	192.168.0.7	07 00 A8 C0
Module Port	2	Module port	20108	8C 4E
Gateway	4	The IP address of gateway	192.168.0.201	C9 00 A8 C0
Work Mode	1	1-TCP client, 0-UOP 2-UDP Server, 3-TCP Server	TCP Client	01
Baud Rate	3	serial port working baud rate	115200	00 C2 01
Serial parameter	1	Data/ Stop/ parity bit	N,8,1	03
Independent ID	3	ID-H,ID-L,ID-type Please fill 0 if don't use	Do not use	00 00 00
Subnet Mask	4	Subnet mask, low in front	255.255.255.0	00 FF FF FF
Sum parity	1	Sum check, from the destination IP to sum parity (including)	Sum check	B9
Complete command string : 55 BA C9 00 A8 C0 2A 20 07 00 A8 C0 8C 4E C9 00 A8 C0 01 00 C2 01 03 00 00 00 00 FF FF FF B9				

Note: Writing in 28 byte but read as 29 byte .The last bit is version number, but it can't write

Note:

1. Except header, Send 28 bytes param when write, return 29 bytes when read. The last byte is firmware version, not writable
2. TCP232-E45/M4 series products, after serial configuration (reload pin back to high level), module will reset automatically, restart after 4s, during this time, do not pull down the Reset pin, or module will restore to factory defaults

4.2. Param table 2

Basic config param as follow (send 55 C5 for write):

name	bytes	description	description	Hex format, low byte in front
header	2	55 C5	header	55 C5
http port	2	Port for webserver, default 80(set this to 0 to disable webserver)	80	50 00
User mac address	6	MAC address (if you doesn't want to change mac, pls write 6 bytes of FF to this area, or read it out then write)	D8 B0 4C 00 01 65	65 01 00 4C B0 D8
Module name	16	Module name, string format, must be ended with 0x00	"USR-TCP232-E45"	55 53 52 2D 54 43 50 32 33 32 2D 45 34 35 00 00 00 ...
Ip type	1	bit7 represent whether use static IP or not, default 0x80 0x80 STATIC IP 0x00 DHCP The other 7 bit should be left default	0x80	80
reserved	3	reserved	Shoule be left default	00 00 00
checksum	1	Checksum, calculated from http port, to the byte befrore checksum, only reserve the low byte.	checksum	xx
Read param Send: 55 c6 Return: 55 C6 50 00 D8 B0 4C 00 1B 3F 55 53 52 2D 54 43 50 32 33 32 2D 45 34 35 00 00 00 00 00 00 E2 E0 Write param Send: 55 C5 50 00 D8 B0 4C 00 1B 3F 55 53 52 2D 54 43 50 32 33 32 2D 45 34 35 00 00 00 00 00 00 fe Return: 4B('K')				