

Socket-Serial Port Configuration API Documentation-v3.2

Time	Version	Remark
01.22.2018	v1.0	Note: This API is currently only applicable to devices with a maximum of two network cards and port 9753.
11.15.2022	v2.2	The connection list and network card configuration have been updated to include a new "Noise" field. Additionally, a new spectrum scanning interface has been added.
11.16.2022.	v2.3	The system information now includes voltage parameters.
11.25.2022	v2.4	The system configuration has added the option to set the network card MAC addresses.
02.01.2023	v2.5	The system configuration now supports custom device models.
03.21.2023	v3.0	The network card configuration has been expanded to include power and mode settings. The network card configuration strategy has been changed to support multiple network cards, while still retaining support for old users using mesh and WiFi mode configurations on port 9753. The data structure on port 9754 has been updated to accommodate multiple network cards, making it compatible with all products.
03.21.2023	v3.0	The device scanning API processes have been integrated.
04.10.2023	v3.1	A new simplified TOPO interface (similar to the WEB display) has been added.
04.11.2023	v3.2	The serial port API has been merged into a unified document. Users can now enable the serial port API switch in advanced settings to utilize API commands through the serial port.

Basic Info

- Communication Types: TCP/Serial Port
- Old TCP Communication Port: 9753 (for legacy mesh and WiFi mode configurations, no longer maintained)
- New TCP Communication Port: 9754

Quick test

- Open the command prompt and enter the following command: telnet 192.168.166.10 9754. This will allow you to access the interactive interface.

Connection Strategy

1. Support one-to-many connections for data transmission.
2. The interaction follows a question-and-answer pattern. The server will first return the following welcome message for TCP communication: "\nconsole: " and then wait for the client to send a command. After sending the data, it will loop to wait for the response. For serial port communication, there is no "console" header information, and the command can be sent directly, waiting for the reply.
3. The data sent is in the format mentioned above + a carriage return ("\n"), and then it waits for the received data.
4. The returned data is in JSON format, and the receiving strategy is to continuously read until a carriage return ("\n") is encountered.

Interactive Parameters

1. **auth JsonData** (Authenticate username and password to obtain a key. The key is valid for 1 hour. No key is required for serial port communication.)
2. **systeminfo JsonData** (Retrieve system basic information and connection details.)
3. **getconfig JsonData** (Retrieve system configuration.)
4. **putconfig JsonData** (Configure system information.)
5. **scanfreq JsonData** (Initiate a spectrum scanning.)
6. **gettopo JsonData** (Retrieve topology data.)
7. **tool JsonData** (Commands for restart and reset.)
8. **exit** (Disconnect the connection. This command is not available for serial port communication.)

Important Notes:

1. Pay attention to the different headers in TCP and serial port communication. TCP will have an additional welcome message to facilitate terminal debugging.
2. For serial port communication, no secret key is required. You can directly send the corresponding commands, and the "secretkey" key-value pair can be completely ignored.

Errorcode Definition

Value	Meaning
0	Normal
10	Authentication failed
11	Unknown command
12	Parameter exception
200	Unknown exception

Data Definition

1. **auth JsonData** (Authenticate username and password to obtain the secret key)

- JsonData definition:

```
{
  "username": "admin", // Username (same as web login)
  "password": "admin" // Password (same as web login)
}
```

- **Auth** response data:

```
{
  "errorcode": "", // Error code
  "secretkey": "21a7510dcea6a28af805c4282bf933b1" // Authentication secret key
}
```

2. **systeminfo JsonData** (Retrieve system basic information)

- JsonData definition:

```
{
  "secretkey": "" // Authentication secret key
}
```

- **systeminfo** response data:

```
{
  "errorcode": "", // Error code (empty if no error)
}
```

```

"systeminfo": {
  "devicename": "", // Mainboard model
  "modelname": "", // Custom device model
  "firversion": "", // Firmware version
  "mac": "", // MAC address
  "runtime": "", // Runtime in seconds
  "temperature": "", // Temperature (example: 48)
  "voltage": "" // Battery voltage (example: 11.524 V)
},
"gps": {
  "uptime": "1565315242", // Timestamp of the last GPS update
  "status": "2", // -1, 0, 1 (-1->GPS module not detected, 0->Data invalid, 1->Data valid)
  "gpsdmy": "020415", // DDMMYY GPS time
  "gpshms": "014002.00", // hhmmss.ss GPS time
  "satellite": "10", // Number of satellites
  "longitude": "114.054112", // GPS longitude
  "latitude": "22.615236", // GPS latitude
  "bdlongitude": "114.054112", // Baidu longitude
  "bdlatitude": "22.615236", // Baidu latitude
  "gglongitude": "114.054112", // Google longitude
  "gglatitude": "22.615236", // Google latitude
  "sirection": "ES", // Southeast (SE, NE, SW, NW-> Southeast, Northeast, Southwest, Northwest)
  "altitude": "105.4", // Altitude (in meters)
  "speed": "0.336", // Ground speed (in km/h)
  "airroute": "0", // Ground track (true north xx degrees)
},
"connectlist": [
  {
    "mac": "", // MAC address
    "ssid": "", // Network
    "noise": "", // Noise
    "signal": "", // Signal
    "rxbit": "", // Receive rate
    "txbit": "", // Transmit rate
  },
  // ... more connectlist items ...
]
}

```

- **systeminfo** sample response data:

```

{
  "errorcode": "0",
  "systeminfo": {
    "mac": "dc5321:c4",
    "runtime": "1448",
    "devicename": "6306",
    "firversion": "V4.6.B(20201028)",
    "temperature": "48",
    "voltage": "11.031 V"
  },
  "gps": {
    "sirection": "-",
    "bdlongitude": "0.0220012958",
    "gpsdmy": "000000",
    "gglatitude": "0.0027284583",
    "airroute": "0.000",
    "speed": "0.000",
    "altitude": "0.000",
    "gglongitude": "0.0154984971",
    "bdlatitude": "0.0087289840",
    "uptime": "0",
    "satellite": "0",
    "status": "0",
    "latitude": "0.0000000000",
    "longitude": "0.0000000000",
    "gpshms": "000000.00"
  },
  "connectlist": [
    {
      "mac": "dc53e6:a2",
      "expect": "36.5",
      "ssid": "mesh1",
      "noise": "-97",
      "rxbit": "86.7",
      "acktimeout": "79S",
      "txbit": "86.7",
      "signal": "-61"
    }
  ]
}

```

3. **getConfig** **JsonData** (Get System Configuration)

- **JsonData** Definition:

```

{
  "secretkey": "" // Authentication Secret Key
}

```

- **getConfig** Return Data:

```

{

```

```

"errorcode": "", // Error code
"device": {
  "name": "", // Device Alias
  "ip": "", // IP Address
  "dhcp": "", // DHCP Allocation (1->Enabled, 0->Disabled)
  "gps": "" // GPS (1->Enabled, 0->Disabled)
},
"wireless": [
  { // Network card list, note that it matches multiple network cards as an array
    "mac": "", // MAC Address
    "wtype": "", // Type (2G, 5G, 2G/5G, xxM)
    "noise": "", // Noise
    "htlist": ["", "", ""], // Bandwidth list (dropdown menu list)
    "htmode": "", // Bandwidth (one value from htlist, representing the current bandwidth)
    "txpowerlist": ["", "", ""], // Power list (dropdown menu list)
    "txpoweradd": "", // Power Offset (actual transmit power is x dBm higher than the set power)
    "txpower": "", // Power (one value from txpowerlist, representing the current power)
    "chlist": ["", "", ""], // Channel list (dropdown menu list)
    "channel": "", // Channel (one value from chlist, representing the current channel)
    "distance": "", // Distance (0 for auto, measured in meters)
    "mode": "", // Mode (ap, sta, mesh, mesh-s <-> Access Point, Client, Mesh Network, Mesh Network-S)
    "wds": "", // WDS Bridge (used for Access Point and Client modes)
    "ssid": "", // SSID Identifier
    "key": "" // Password (empty if no encryption)
  },
  {} // ... additional entries if available
]
}

```

• **getConfig** returns sample data:

```

{
  "device": {
    "dhcp": "0",
    "gps": "0",
    "name": "6306",
    "ip": "192.168.166.10"
  },
  "errorcode": "0",
  "wireless": [
    {
      "mac": "dc53d8:d0",
      "noise": "-102",
      "ssid": "MeshAP0",
      "distance": "auto",
      "txpower": "20",
      "wds": "1",
      "htlist": ["HT20", "HT40"],
      "txpoweradd": "0",
      "key": "",
      "channel": "4-2427",
      "chlist": [
        "auto", "1-2412", "2-2417", "3-2422", "4-2427", "5-2432", "6-2437", "7-2442", "8-2447", "9-2452", "10-2457", "11-2462", "12-2467", "13-2472", "14-2484"
      ],
      "mode": "ap",
      "txpowerlist": [
        "auto", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26"
      ],
      "wtype": "2G"
    },
    {
      "mac": "dc53c9:b5",
      "noise": "-100",
      "ssid": "MeshAP1",
      "distance": "2000",
      "txpower": "auto",
      "wds": "0",
      "htlist": ["HT2", "HT5", "HT10", "HT20", "HT40"],
      "txpoweradd": "6",
      "key": "",
      "channel": "106-1430",
      "chlist": [
        "auto", "100-1400", "101-1405", "102-1410", "103-1415", "104-1420", "105-1425", "106-1430", "107-1435", "108-1440", "109-1445", "110-1450", "111-1455", "112-1460"
      ],
      "mode": "mesh",
      "txpowerlist": [
        "auto", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30"
      ],
      "wtype": "1400M"
    }
  ]
}

```

4. **putconfig** **JsonData** (Configuration of System Information)

• **JsonData** definition:

```

{
  "secretkey": "", // Secret Key
  "device": {
    "name": "", // Device Alias

```

```

"ip": "", // IP Address
"dhcp": "", // DHCP Allocation (1->Enable, 0->Disable)
"gps": "" // GPS (1->Enable, 0->Disable)
},
"wireless": [{
"mac": "", // MAC Address
"htmode": "", // Bandwidth (one of the values in htlist, representing current bandwidth)
"txpower": "", // Power (one of the values in txpowerlist, representing current power)
"channel": "", // Channel (one of the values in chlist, representing current channel)
"distance": "", // Distance (0 for auto, measured in meters)
"mode": "", // Mode (ap, sta, mesh, mesh-s <-> Access Point, Client, Smart Network, Smart Network-S)
"wds": "", // WDS Bridging (0->Disable, 1->Enable, used in Access Point and Client modes)
"ssid": "", // SSID Identifier
"key": "" // Password (empty if no encryption)
}, {} ...]
}

```

- **putconfig** response data:

```

{
"errorcode": "", // Error Code
}

```

5. **scanfreq** JsonData (Requesting Spectrum Scan)

- JsonData definition:

```

{
"secretkey": "", // Authentication Secret Key
"mac": "" // Network Interface MAC Address
}

```

- **scanfreq** response data:

```

{
"errorcode": "", // Error Code
"result": [{
"freq": 1457.8125, // Frequency
"max": -100.405815, // Maximum Signal
"avg": -107.917779, // Average Signal
"min": -131.898956 // Minimum Signal
}], {}
}

```

6. **gettopo** JsonData (Requesting Topology Data)

- JsonData definition:

```

{
"secretkey": "" // Authentication Secret Key
}

```

- **gettopo** response data:

```

{
"terminal": [{ // Device node
"status": "", // Status (0->Offline, other values represent online status)
"alarm": "", // Alarm (displayed if not empty)
"icon": "", // ap, sta, mesh, route, cloud (icon display)
"device": "", // Mainboard model
"name": "", // Alias
"lan": {
"mac": "", // MAC Address
"ip": "", // IP Address
},
"wlanlist": [{ // Wireless card 1, wireless card 2, ... (label switching)
"mac": "", // MAC Address
"wtype": "", // Type
"mode": "", // Mode
"config": {
"channel": "", // Channel
"ssid": "", // SSID
}, // Display -> mac, wtype, mode,
channel, ssid
"connectlist": [{ // Connection list
"mac": "", // MAC Address
"signal": "", // Signal strength
"rxrate": "", // Downlink rate
"txrate": "", // Uplink rate
}, {}, {} ...],
}, {}, ...],
"location": { // Topology map coordinates
"x": "", // x-coordinate (random if 0)
"y": "" // y-coordinate (random if 0)
},
"gps": { // Online map coordinates
"uptime": "1565315242", // Update timestamp
"status": "2", // -1,0,1 (-1->GPS module not detected, 0->
data invalid, 1->data valid)
"gpsdmy": "020415", // DDMMYY GPS time
"gpshms": "014002.00", // hhmmss.ss GPS time
"satellite": "10", // Number of satellites
"longitude": "114.054112", // GPS longitude
"latitude": "22.615236", // GPS latitude
"bdlongitude": "114.054112", // Baidu longitude

```

```

"bdlatitude": "22.615236", // Baidu latitude
"gglongitude": "114.054112", // Google longitude
"gglatitude": "22.615236", // Google latitude
"sirection": "ES", // East-Southeast (SE,NE,SW,NW-> Southeast, Northeast, Southwest, Northwest)
"altitude": "105.4", // Altitude (in meters)
"speed": "0.336", // Ground speed (in km/h)
"airroute": "0", // Ground track (degrees north)
}
} ... {} ... ],
"line": [{ // Device connections (using MAC addresses from wlanlist)
"style": "", // 0->Solid line, 1->Curved line
"color": "", // Color (green, yellow, red)
"maca": "", // MAC Address (connection)
"macb": "", // MAC Address (connection)
"signal": "", // Signal strength (used for signal filtering)
"text": "", // Display text a->b (direct display)
"text2": "", // Display text b->a (direct display)
} ... {} ... ]
}

```

- Example of **gettopo** response data:

```

{
"terminal": [
{
"status": "111",
"alarm": "",
"device": "6308",
"name": "SmartMesh",
"icon": "mesh",
"lan": {
"mac": "dc5327:3c",
"ip": "192.168.166.12"
},
"wlanlist": [
{
"mode": "ap",
"mac": "dc5327:3b",
"wtype": "2G",
"config": {
"channel": "11-2462",
"ssid": "Mesh-AP0"
},
"connectlist": []
},
{
"mode": "mesh",
"mac": "dc5327:3a",
"wtype": "5G",
"config": {
"channel": "36-5180",
"ssid": "Mesh-AP1-SS"
},
"connectlist": [
{
"mac": "dc5326:e6",
"signal": "-84",
"rxrate": "6.0",
"txrate": "180.0"
},
{
"mac": "dc53db:43",
"signal": "-46",
"rxrate": "390.0",
"txrate": "6.0"
}
]
},
],
"location": {
"x": "0",
"y": "0"
},
},
{
"status": "101",
"alarm": "",
"device": "6308",
"name": "SmartMesh",
"icon": "mesh",
"lan": {
"mac": "dc5326:e8",
"ip": "192.168.166.10"
},
"wlanlist": [
{
"mode": "ap",
"mac": "dc5326:e7",
"wtype": "2G",
"config": {
"channel": "11-2462",

```

```
"ssid": "Mesh-AP0"
},
"connectlist": [
{
"mac": "4c9cf4:3b",
"signal": "-75",
"rxrate": "6.0",
"txrate": "1.0"
}
],
{
"mode": "mesh",
"mac": "dc5326:e6",
"wtype": "5G",
"config": {
"channel": "36-5180",
"ssid": "Mesh-AP1-SS"
},
"connectlist": [
{
"mac": "dc53db:43",
"signal": "-62",
"rxrate": "390.0",
"txrate": "260.0"
},
{
"mac": "dc5327:3a",
"signal": "-87",
"rxrate": "0",
"txrate": "390.0"
}
]
},
"location": {
"x": "0",
"y": "0"
},
{
"status": "111",
"alarm": "",
"device": "6308",
"name": "SmartMesh",
"icon": "mesh",
"lan": {
"mac": "dc53db:45",
"ip": "192.168.166.11"
},
"wlanlist": [
{
"mode": "ap",
"mac": "dc53db:44",
"wtype": "2G",
"config": {
"channel": "11-2462",
"ssid": "Mesh-AP0"
},
"connectlist": []
},
{
"mode": "mesh",
"mac": "dc53db:43",
"wtype": "5G",
"config": {
"channel": "36-5180",
"ssid": "Mesh-AP1-SS"
},
"connectlist": [
{
"mac": "dc5326:e6",
"signal": "-64",
"rxrate": "260.0",
"txrate": "390.0"
},
{
"mac": "dc5327:3a",
"signal": "-50",
"rxrate": "292.6",
"txrate": "325.0"
}
]
},
"location": {
"x": "0",
"y": "0"
}
},
],
```

```

"line": [
{
"style": "0",
"color": "red",
"signal": "-85",
"maca": "dc5327:3a",
"macb": "dc5326:e6",
"text": "-85/-87/6.0/180.0",
"text2": "-87/-85/0/390.0"
},
{
"style": "0",
"color": "green",
"signal": "-48",
"maca": "dc5327:3a",
"macb": "dc53db:43",
"text": "-48/-50/390.0/6.0",
"text2": "-50/-48/292.6/325.0"
},
{
"style": "0",
"color": "green",
"signal": "-63",
"maca": "dc5326:e6",
"macb": "dc53db:43",
"text": "-63/-64/390.0/260.0",
"text2": "-64/-63/260.0/390.0"
}
]
}

```

7. tool JsonData (Restart/Reset Command)

- JsonData definition:

```

{
"secretkey": "", // Secret Key
"action": "" // "reboot"->Restart, "reset"->Restore default configuration
}

```

- tool response data:

```

{
"errorcode": "", // Error Code
}

```

Device Scanning API Process

- UDP broadcast sender port: 60083
- UDP broadcast receiver port: 60084

User program listens on port 60084, then broadcasts data to port 60083. Port 60084 will receive replies from all devices.

- Sending data structure:

```

{
"time": "0", // Identifier, will be returned, can be used to verify data validity
"key": "12345678" // Secret key, must be "12345678"
}

```

- Example data:

```

{"time": "0", "key": "12345678"}

```

- Receiving data structure:

```

{
"time": "0", // Identifier, usually a timestamp
"device": "6306", // Model
"devicename": "SmartMesh", // Alias
"version": "V3.5.Q(20220330)", // Version number
"ethernet": "DC53F3:00", // Ethernet MAC address
"wlan": ["DC53F2:FF", "DC53F2:FE"], // WLAN MAC addresses
"lan": "192.168.166.10", // Device IP address
"wan": "0.0.0.0" // External IP address
}

```

- Example data:

```

{"time": "0", "device": "6306", "devicename": "SmartMesh", "uptime": "419", "version": "V3.M.2(20221212)", "ethernet": "DC53D2:99", "wlan": ["DC53D9:04", "DC53C9:CC"], "lan": "192.168.166.10", "wan": "0.0.0.0"}

```