****

**EP -W100 Manual Applicable Products**

|  |  |
| --- | --- |
| **model** | **Product Description** |
| EP - W100-485 | RS485 serial port to WiFi |
| EP - W100-232 | RS232 Serial to WiFi |
| EP -W100 -TTL | TTL Serial to WiFi |

**Features**

* Serial port (optional RS232/RS485/TTL) and WiFi data conversion
* Dual-core 240MHz high-performance MCU
* 150Mbps wireless rate
* 5~36V wide voltage power supply
* Dual watchdog protection
* Each serial port supports dual-channel Socket communication
* Support TCP/UDP client, server, HTTP client, MQTT
* Supports connection with Alibaba Cloud, Baidu Cloud, and Mobile Onenet
* Support ModbusTCP/RTU protocol conversion
* Support batch export and import of parameters
* Built-in web page
* Support web page upgrade and remote upgrade

**Order record**

[Chapter 1 Product Introduction 4](#_Toc25935)

[1.1 Overview 4](#_Toc26409)

[1.2 Specifications 5](#_Toc26403)

[1.3 Product Dimensions /Interface Diagram 6](#_Toc13646)

[1.4 Quick Start 7](#_Toc1141)

[1.4.1 Hardware Connection 7](#_Toc15930)

[1.4.2 Connect the device to WiFi and log in to the device's built-in web page 8](#_Toc24236)

[1.4.3 Setting Socket Parameters 8](#_Toc2498)

[1.4.4 Test Communication 9](#_Toc16944)

[Chapter 2 Product Function 10](#_Toc3848)

[2.1 WiFi 10](#_Toc7302)

[1. AP mode does not support scanning hotspots 11](#_Toc21671)

[2.2 Serial port 11](#_Toc5222)

[2.2.1 Serial port parameters 11](#_Toc16112)

[2.2.2 Packaging Mechanism 11](#_Toc23404)

[2.2.3 Serial port working mode 12](#_Toc25950)

[2.3 Data Transmission 13](#_Toc20040)

[2.3.1 TCP Client/Server 13](#_Toc20595)

[2.3.2 UDP Client/Server 15](#_Toc24491)

[2.3.3 HTTP Client 16](#_Toc19342)

[2.3.4 Point-to-point transmission 18](#_Toc27443)

[2.3.5 MQTT 19](#_Toc4745)

[2.3.6 Heartbeat Packet, Registration Packet 20](#_Toc14633)

[2.4 Accessibility 21](#_Toc8470)

[2.4.1 Internet Search 21](#_Toc212)

[2.4.2 Network AT 22](#_Toc23095)

[2.4.3 Firmware Upgrade 22](#_Toc26278)

[2.5 Exception Handling 24](#_Toc12358)

[Chapter 3 Parameter Setting 25](#_Toc7312)

[3.1 Web page settings 25](#_Toc31331)

[3.1.1 Status Page 25](#_Toc15938)

[3.1.2 System Management Page 25](#_Toc19022)

[3.1.3 Port Settings Page 26](#_Toc4131)

[3.2 AT Command 27](#_Toc28136)

[3.2.1 AT command format 27](#_Toc16191)

[3.2.2 AT command error code information 28](#_Toc25078)

[3.2.3 Quickly Master AT Commands 28](#_Toc29063)

[3.2.4 Detailed explanation of AT commands 29](#_Toc26598)

[Disclaimer 44](#_Toc12553)

[Contact 45](#_Toc16016)

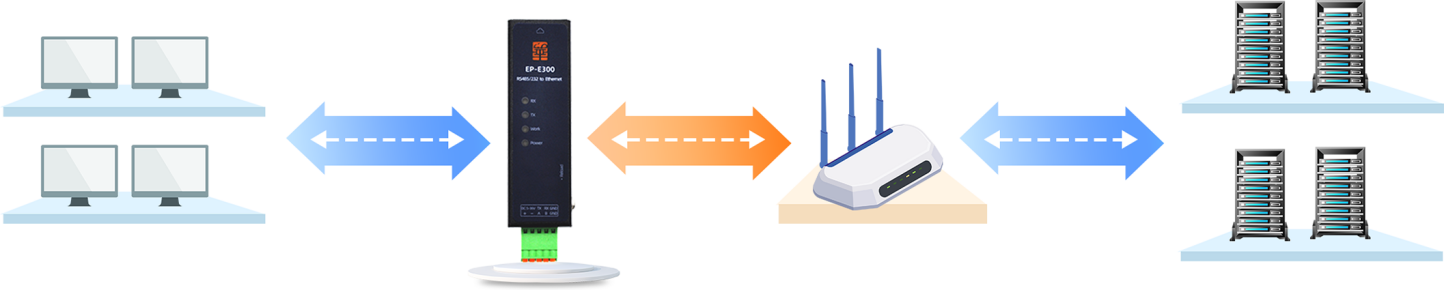
# Product Introduction

## Overview

EP -W100 series products are data terminals that can realize serial port and WiFi transmission. They use dual-core high-performance MCU with a main frequency of up to 240 MHz, support wide voltage (5 ~ 36 V) power supply, and can select RS 485/RS232/TTL for serial port . Each serial port corresponds to two sockets, and supports TCP /UDP / HTTP/MQTT/ point-to-point and other transmission protocols.

Excellent hardware performance and rich software functions enable this product to be applied in various industries and complex scenarios, such as power grid, transportation, fire protection, industrial production, meteorological environment, agriculture and forestry, mining, etc.

Typical application of the product is shown in Figure 1 :



User equipment EP-W100 Router or switch Device server

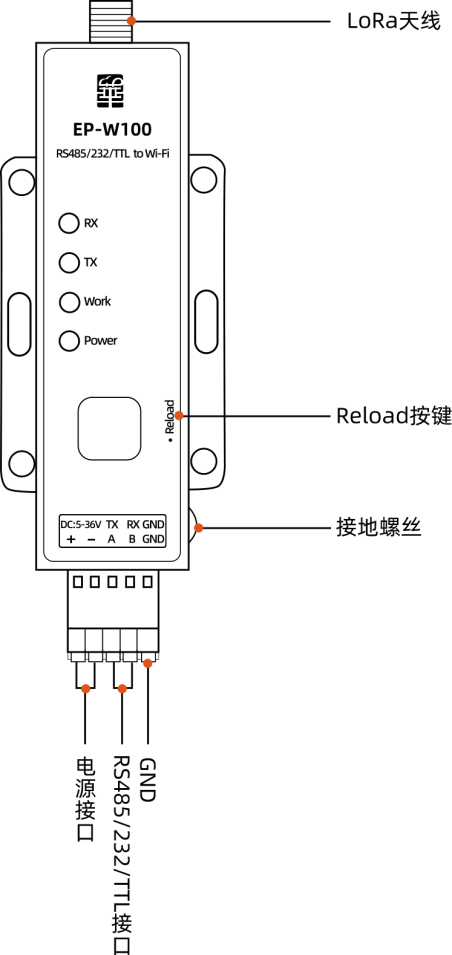
Figure 1 Network topology diagram

## Specifications

Table 1 Specifications

|  |  |  |
| --- | --- | --- |
|  | **entry** | **parameter** |
| Electrical parameters | size | 74.6×22.6×16mm (excluding side ears and terminals) |
| Operating temperature | -40 ~ 85 ℃ |
| Operating humidity | 5 % ~ 95% |
| Power supply interface | 2.54\*2 P terminal |
| Supply voltage | 5 ~ 36 V |
| Supply current | 55 mA @12V |
| Wireless parameters | standard | 8 02.11 B / G / N @ 2.4 GHz |
| model | AP (can connect 1 to 6 STAs) /STA/APSTA |
| Transmit power | 20.5dBm Max |
| Antenna interface | SMA antenna base (external screw and internal hole) |
| Encryption Type | WPA-PSK/WPA2-PSK/WPA3-PSK |
| Serial port (2 channels) | Interface Type | 2.54 terminal​ |
| Baud rate | 1200 ~ 4 60800 bps |
| Check digit | NONE /ODD/EVEN |
| Data bits | 7/ 8 |
| Stop bits | 1/2 |
| Hardware flow control | none |
| LED​ | Power | Power indicator |
| Work | Working status indicator  Slow flashing means WiFi is connected, fast flashing means WiFi is disconnected |
| T X | Serial Port T x Indicator |
| R​ | Serial Port Rx indicator |
| button | Reload | Press and hold for more than 3 seconds to restore default parameters |
| Software Features | Network Protocol | DHCP/DNS/TCP/UDP / HTTP /ICMP / MQTT |
| Transparent transmission protocol | TCP Client/Server (Server is limited to a maximum of 8 clients)  U DP Client/Server  QT​  H TTP |
| Accessibility | Built-in web page | Support parameter setting and firmware upgrade |
| AT commands | Serial AT / Network AT |
| LAN Search | Search for devices and execute network AT commands |
| Equipment maintenance | Serial port upgrade, network upgrade, exception handling |

## Product size drawing /interface drawing

Unit: mm

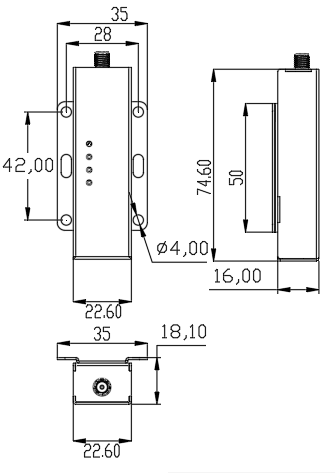


Figure 2 Dimensional drawing /interface drawing

Default serial port parameters:

|  |  |  |
| --- | --- | --- |
| **Serial Port** | **Options** | **parameter** |
| RS 232/RS485 | Baud rate | 1 15200 |
| Check digit | NONE |
| Data bits | 8 |
| Stop bits | 1 |

## Get started quickly

This section introduces how to quickly use this product to achieve basic transparent transmission functions. The final effect is to achieve data transmission between the device as a TCP server and the TCP client on the computer. The overall process is divided into the following steps:

1. Hardware Hookup
2. Connect the device to WiFi and log in to the device's built-in web page
3. Configure the device socket parameters and restart
4. Test communication

Note: If the device WiFi is working in non-AP or non-APSTA mode, please press and hold the Reload button for 5 seconds before operating.

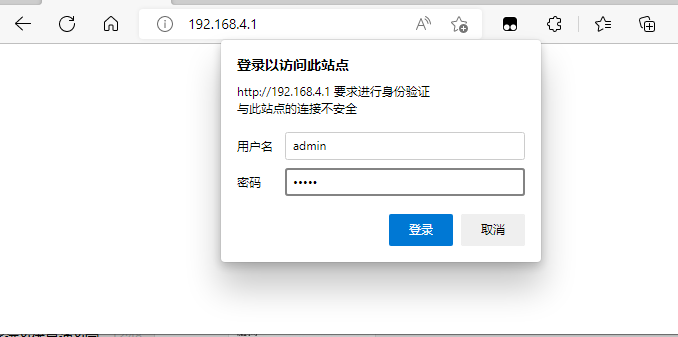
### Hardware Hookup

1. Use a USB to serial cable to connect the computer and the device
2. Power on the device

### Connect the device to WiFi and log in to the device's built-in web page

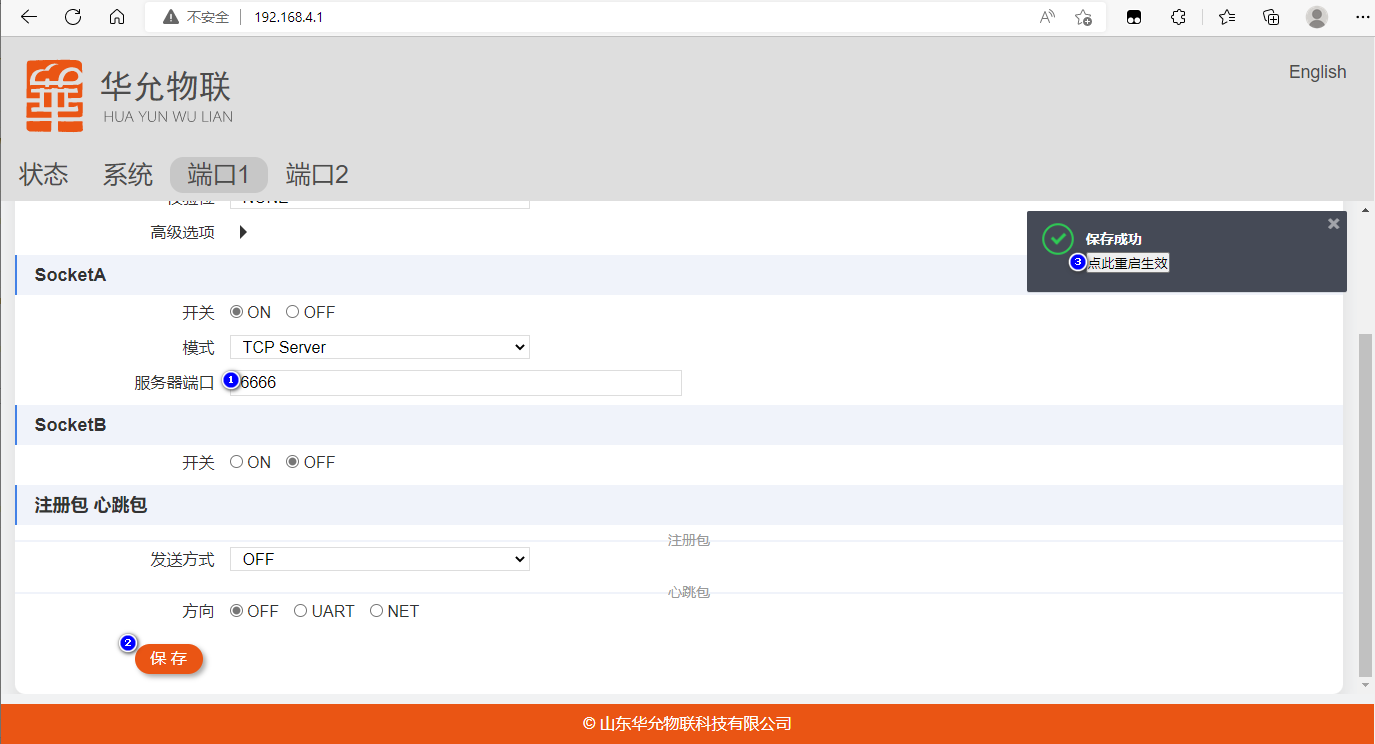


searching for the HY -W100 hotspot, connect it, start the browser and open the device's built-in web page 1 92.168.4.1 . Enter the username and password in the pop-up window (both are admin by default) and click OK to enter the web page



### Setting Socket Parameters

Enter **the Port 1** page, set the SocketA parameter to TCP Server port 6 666. Enter the Device Port page, and then click the **Save** button and the Restart button



### Test communication

After the device restarts, the computer reconnects to the device hotspot, and then starts the network and serial port debugging assistant.

The network debugging assistant uses TCP Client to connect to 1 92.168.4.1:6666 , and the serial port debugging assistant uses the 1 15200/8/1/NONE parameter to open the serial port and test data sending and receiving:



Figure 8 Data transmission and reception test

# Product Features

The main functional block diagram of the product is as follows:

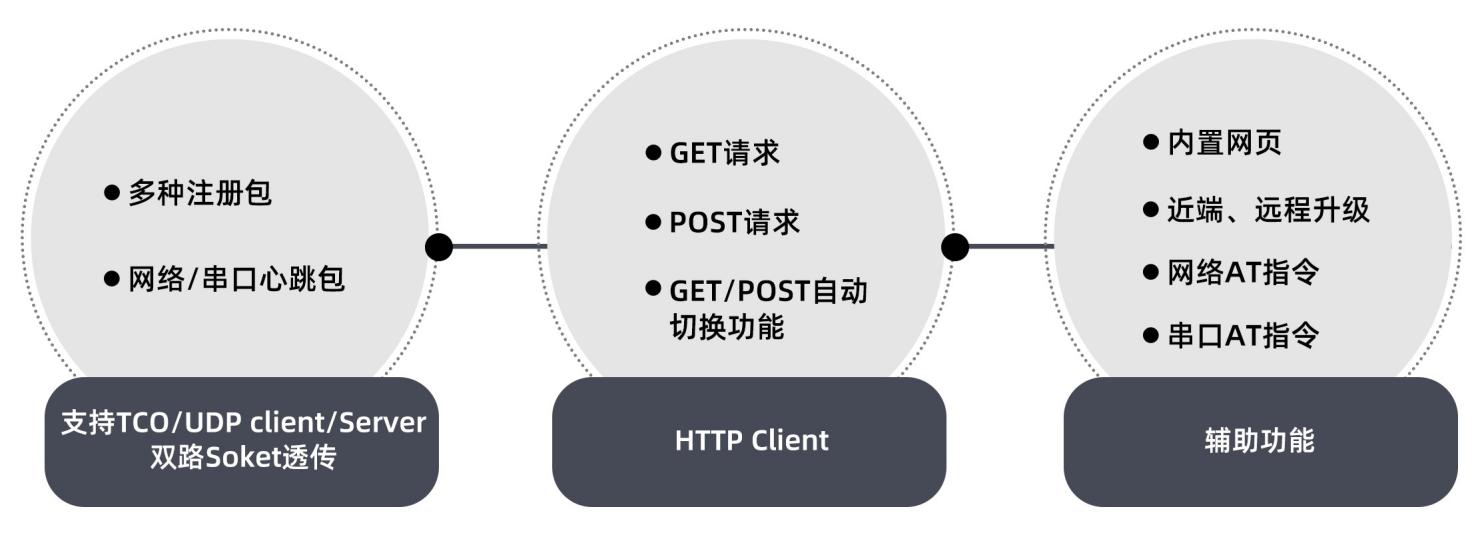


Fig. 9

## Wi -Fi

WiFi supports three modes: AP, STA, and APSTA ( default ) :

AP mode: The device sends out wireless signals, which can be scanned and connected by mobile phones, computers and other devices. The default IP in AP mode is 1 92.168.4.1 . Through this IP, you can open the device's built-in web page for parameter configuration or Socket communication.

STA mode: The device can be connected to a router. After being connected to the router, the device can perform Socket communication with LAN or extranet devices.

AP + STA mode: The device can send signals and access the router at the same time, which is equivalent to AP + STA . In this mode, the wireless channel sent by the device is not the user-set value, and it will automatically switch to the router channel accessed by STA.

Notice:

1. Scanning hotspots is not supported in AP mode

2. Users select AP or STA mode according to the scenario. AP + STA will reduce wireless performance.

## Serial Port

The product supports 1 serial port. If the letter n is included in the AT commands mentioned below , it means that the parameters set by the command are related to the serial port No. n of the device .

### Serial port parameters

Table 2 Serial port parameters

|  |  |  |
| --- | --- | --- |
| **Serial Port** | **Options** | **parameter** |
| RS 232/ RS485 /TTL | Working Mode | Instruction Mode |
| Transparent mode (default) |
| Baud rate | 1200 ~ 4 60800 bps (default 1 15200 ) |
| Check digit | NONE /ODD/EVEN (default NONE) |
| Data bits | 8 (Default 8) |
| Stop bits | 1/2 (default 1) |

### Packaging mechanism

In order to improve the network transmission performance, the serial port receives data and packages it into a frame of data before forwarding it to the network. This product supports two ways of packaging: data length or data receiving interval. As long as one of the two methods is met, data will be forwarded. The two packaging conditions are as follows:

* Data length packaging: data length ≥ packaging length (default 1 024 , support 8 ~ 1460 )
* Data interval packaging: adjacent character interval ≥ packaging interval (support 0 ~ 300ms , 0 means adaptive with baud rate, default 0)

Command example: AT+UARTTL 1=30,1024 sets the packet time to 30ms and the length to 1024

Note: Data may be concatenated under the TCP protocol. If you have strict requirements on the length of data packets, be sure to add a mechanism for splitting packets at the application layer.

### Serial port working mode

The serial port supports two working modes:

* AT command mode: The data received by the serial port is executed as a command, and the data sent by the network is discarded. Parameters can be queried and set in command mode.
* Transparent transmission mode: After receiving data, the serial port forwards it through the Socket. It is the default mode when the device is turned on.

Users can query and set the parameters of this product by sending AT commands through the serial port communication of a computer or MCU. Each command line can only contain one AT command, and a single command has a maximum of 256 bytes . After setting new parameters, they are automatically saved and take effect after restart .

#### Enter command mode

The timing of entering AT mode is shown in Figure 1 0 Among them, "UART " represents the user serial port device, and "Huayun Terminal" represents this series of products.

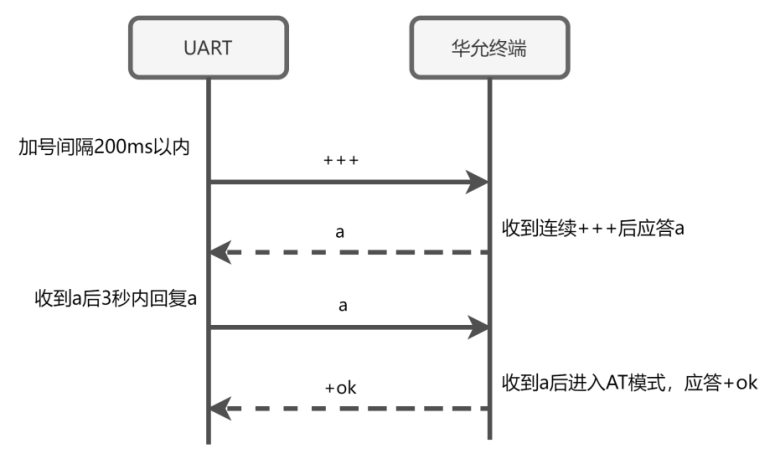


Fig.10​

If the action of sending the entry command is aborted, the data sent during the period will be forwarded to the Socket. If the user serial port receives "+ok", it means that the AT command mode has been successfully entered. The command mode will not be saved when the power is off.

In order to reduce the interference of data sent from the network on the above process, this product will suspend network data output for up to 3 seconds after receiving + ++ . Even so, 'a' may still be mixed with other data. The recommended method is:

1. device sends +++
2. a directly after a delay (greater than the packaging time, less than 3 seconds, 500 ms is recommended )
3. The user device determines whether it has received +ok ( remember to receive + ok before sending or receiving commands )

For detailed description of AT commands, please refer to the AT commands section.

#### Exit command mode

In AT command mode, you can switch to transparent transmission mode by using AT command AT + EXIT or restarting the device. Note that the command ends with the escape character \ r\n .

## Data Transfer

Each serial port of this series of products supports two Socket links. By default, only one Socket is enabled for each serial port. When two Sockets are enabled at the same time, the data received by the serial port will be forwarded to the two Sockets respectively; and when the two Sockets receive data, they will be output in sequence from the same serial port.

When the socket is used as a client, the local port uses a random port by default and can also be set to a fixed port. Please note that the same port cannot be used when the protocols are the same.

### TCP Client/Server

The TCP protocol is a connection-oriented reliable transmission protocol. It is recommended to use the TCP protocol in scenarios where data integrity is critical.

The TCP protocol is a C /S architecture. The server and client need to establish a connection before data can be exchanged. The communication process is: the server passively listens to a port, and the client can actively initiate a connection request to the server. After the connection is established, both parties can send data to each other. This series of products will enable the Keepalive detection function when using the TCP protocol, which can effectively avoid dead links.

#### TCP Client

Enable TCP When the device is connected to the client, the client will automatically initiate a connection to the server after obtaining the IP. If the connection fails or the port fails, this product will automatically initiate a reconnection with a reconnection interval of 1 second.

Setting example, using TCP of Socket B of serial port 3 Client communication:

1. Open Socket 3 B: AT+SOCKEN3B=ON
2. Set the socket parameters:

AT+SOCK3B=TCPC,192.168.1.16,7788

1. Reboot to take effect: AT+REBOOT

After restarting, the device automatically connects to the TCP server at 192.168.1.16:7788 . After the connection is successful, serial port 3 can exchange data with the server.

#### TCP Server​

When the device acts as a server, it will listen to the port set by the user, and establish a connection with the client when receiving a client access request. Currently, each server is limited to connecting to a maximum of 3 clients. If the number of clients exceeds the limit, the server will kick the earliest connected client offline and connect a new client. When the server connects to multiple clients, the data received by the serial port will be forwarded to each client.

Setting example, using TCP of Socket A of serial port 1 Server communication:

1. Open Socket 1 A: AT+SOCKEN1A=ON
2. Set Socket parameters: AT+SOCK1A=TCPS,192.168.1.16,7788
3. Reboot to take effect: AT+REBOOT

After restarting the device and connecting to the Client, data can be sent between serial port 1 and the Client.

### UDP Client/Server

The UDP protocol is a connectionless transport layer protocol that provides unreliable information transmission services, but UDP also has its advantages: because it does not require connection and transmission control, UDP can achieve more efficient data transmission and does not require additional resources to maintain connections.

Strictly speaking, the UDP data sender and receiver are completely equal, and there is no need to distinguish between Client/ Server or establish a connection. One party only needs to know the other party's IP and port to send data to the other party. In order to facilitate users to use this product, UDP is artificially divided into two modes: Client and Server.

#### UDP Client

UDP In Client mode, the target IP and port remain unchanged, and data can be sent to a fixed target. In UDP mode, it is recommended to fix the local port, otherwise the device needs to send a packet of data to the other end before the other end can send data to the device.

Setting example, using UDP Client communication of Socket B of serial port 1 , local port is 5678 , and remote UDP server is 192.168.1.16:7788 :

1. Open Socket 1 B: AT+SOCKEN1B=ON
2. Set Socket parameters: AT+SOCK1B=UDPC,192.168.1.16,7788,5678
3. Reboot to take effect: AT+REBOOT

#### U DP Server

UDP In Server mode, the local port set by the user is bound. When receiving data from a certain "Client", the Server will use this "Client" as the sending target. In other words, in Server mode, data sent by any number of "Clients" can be received, and the sending target is the single "Client" that has communicated recently.

Setting example, using UDP Server communication of Socket B of serial port 1 :

1. Open Socket 1 B: AT+SOCKEN1B=ON
2. Set Socket parameters: AT+SOCK1B=UDPS,192.168.1.16,7788
3. Reboot to take effect: AT+REBOOT

After restarting, the device starts listening to UDP port 7788. The device does not send data to the target before receiving it. After receiving the data from the Client, the device will take the Client as the target, and the two parties can now transmit data to each other.

### H TTP Client

HTTP is a simple request-response protocol. In simple terms, HTTP is a request sent by the client and then responded by the server. We usually use the HTTP protocol to browse the web.

The protocol versions supported by this product are HTTP 1.0 and HTTP 1.1 , and can implement HTTP GET / POST requests. This product supports AUTO mode, allowing users to flexibly select the request method when initiating HTTP requests. The following examples illustrate the use of GET / POST / AUTO .

Test interface description: -

* GET interface: <http://www.rt-thread.com/service/rt-thread.txt>A successful request returns the rt -thread introduction.
* POS interface: <http://www.rt-thread.com/service/echo>After submitting the data, the same data will be automatically returned.

Note: HTTP and Socket share the same address and port parameters. Before using HTTP, make sure that the corresponding Socket function has been enabled. HTTP :// in the address can be omitted.

#### GET request

1. Switch HTTPC mode and set the address and port AT+SOCKnA=HTPC,www.rt-thread.com,80
2. Set the packet header AT+HTPHD n =Connection: keep-alive (If there are multiple packet headers, separate them with “|”)
3. Set GET request mode AT+HTPREQ n =GET
4. Reboot to take effect AT+REBOOT
5. After the serial port n sends /service/rt-thread.txt request successfully, the server returns r t-thread introduction

Note: The URL set by the AT+HTPURL n command is only valid for POST requests.

#### POST request

1. Switch HTTPC mode and set the address and port AT+SOCK n A=HTPC,www.rt-thread.com,80
2. Set URL AT+HTPURL n =/service/echo
3. Set the packet header AT+HTPHD n =Connection: keep-alive (If there are multiple packet headers, separate them with “|”)
4. Set POST request mode AT+HTPREQ n =POST
5. Reboot to take effect AT+REBOOT
6. Serial port n sends any data, and the server automatically replies with the same data

#### A UTO

In AUTO mode, users only need to switch to HTTPC mode and set the packet header. Other parameters such as server address, port, URL, request method, data, etc. can be flexibly changed when sending data.

1. Switch to HTTPC mode and AT+SOCK n A=HTPC,null,80 (address and port can be set arbitrarily)
2. Set the packet header AT+HTPHD n =Connection: keep-alive (If there are multiple packet headers, separate them with “|”)
3. Set AUTO request mode AT+HTPREQ n =AUTO
4. Reboot to take effect AT+REBOOT

**GET request** : Serial port n can send a complete URI, such as any of the following formats (when no port is specified, port 80 is used by default)

* http://www.rt-thread.com/service/rt-thread.txt
* http://www.rt-thread.com:80/service/rt-thread.txt
* www.rt-thread.com/service/rt-thread.txt

**POST request format** : URI and data n are separated by carriage return and line feed escape characters \r\n , such as any of the following formats:

* http://www.rt-thread.com/service/echo\r\nTEST
* http://www.rt-thread.com:80/service/echo\r\nTEST
* [www.rt-thread.com/service/echo\r\nTEST](http://www.rt-thread.com/service/echo/r/nTEST)

### Point-to-point transmission

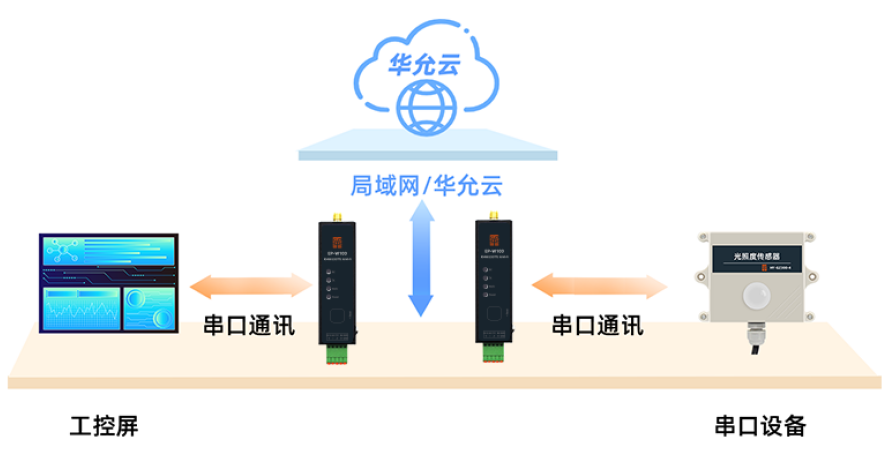


Figure 7 Point-to-point transmission network topology diagram

Point-to-point can realize data transmission between devices, and users can monitor data and remotely manage and maintain the device in the cloud. Our point-to-point function supports one-to-one, one-to-many, many-to-one and many-to-many. In addition, different series of products such as Ethernet and 4G DTU can also realize point-to-point transmission.

Point-to-point function flow and examples:

1. Contact our customer service to obtain the ID number (one number for each device). Assume that device 1 is 111 and device 2 is 222.
2. Configuring Device 1

* Configure the socket to point-to-point mode: AT+ SOCK1A=EDP, huayuniot ,888
* Configuration sets the point-to-point target ID to the ID of device 2: A T+EDP1=222,OFF
* Save and Restart

1. To operate device 2, the process is the same as that of device 1. You need to change the target ID to 1 11 of device 1. After the operation is completed, device 1 and device 2 can send data to each other.

### MQTT

MQTT is a "lightweight" communication protocol based on the publish/subscribe model. It uses TCP/IP to provide network connection, can implement message shielding transmission for payload content, has low overhead, and can effectively reduce network traffic.

The MQTT function of the DTU device is a quasi-transparent transmission mechanism. When a connection is established with the server, it will automatically subscribe to the preset topic and preset a topic for publishing. The DTU internally handles MQTT subscription and publishing transactions, and the user terminal device only needs to receive and send message content.

There are many MQTT parameters, users can set them according to their own needs. For more detailed parameters, please refer to the relevant AT commands

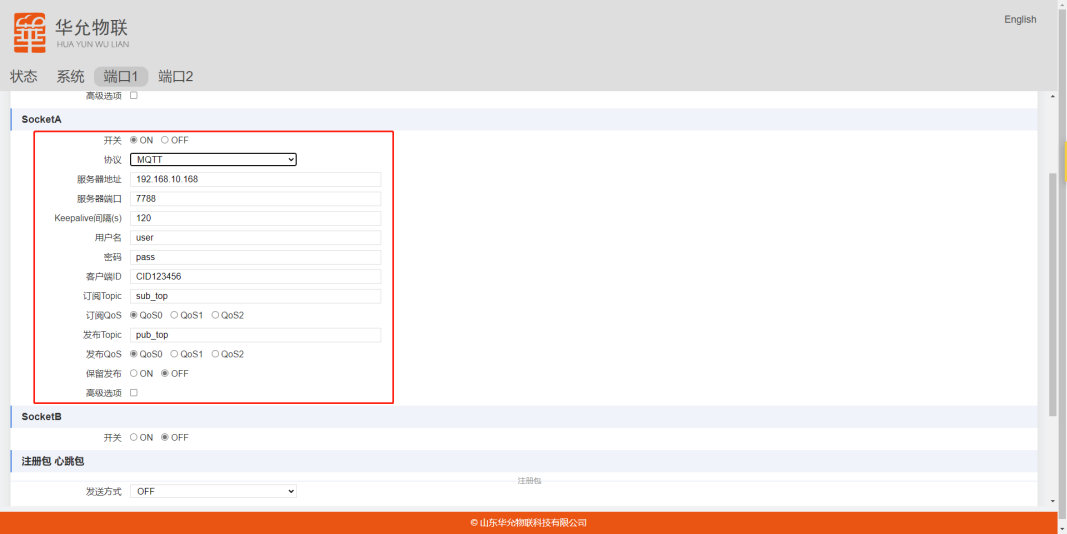


Figure 8 MQTT parameter settings

### Heartbeat packet, registration packet

Heartbeat packet and registration packet are additional functions of data transparent transmission. Each serial port can set the heartbeat packet function and registration packet function independently, which are closed by default.

#### Heartbeat Packet

The heartbeat packet function is a keep-alive mechanism on the application layer that can periodically send data to the network or serial port to facilitate users to confirm whether the device is working properly.

* Network heartbeat packet supports TCP /UDP Client mode
* The serial port heartbeat packet is not limited by the network mode. When the device enters the AT command state, the heartbeat packet will stop sending.

Setting example: Enable serial heartbeat packet of serial port 2 and send a packet of data every 30 seconds:

1. Serial port 2 heartbeat mode is set to serial port: AT+HEART MD 2=UART
2. Set the heartbeat interval: A T+HEARTTM2=30
3. Reboot to take effect: AT+REBOOT

#### Registration Package

The function of the registration package is that when the device communicates with the user server, the device actively sends some specific data, and the server can distinguish the client based on this. The registration package supports TCP /UDP Client and can be set to three sending methods:

* First send: TCP Client reports after each successful connection; UDP Client only reports once after connecting to the Internet
* Data carrying: sent as a data packet header and data at the same time
* First send + data carrying

Setting example: MAC heartbeat of serial port 2 is sent only once:

1. Use MAC registration package: AT+ REG TP2=MAC
2. Open the registration package, only send it for the first time: A T+ REGM D2= FIRS T
3. Reboot to take effect: AT+REBOOT

## Accessibility

### Web Search

The network search function makes it easy for users to maintain devices in the LAN. Users can search the list of devices in the LAN and send AT commands to the devices.

broadcast the search keyword HuayunIOT to port 8168. After receiving the specified broadcast packet, the device will reply with its own IP , MAC, model, and version .

For example: 192.168.1.4,0080E1134527, HY -E TH ,V1.0.0



Fig.10​ Web Search

Modify the port and keyword command: AT+SEARCH =8168, HuayunIOT

### Network AT

Devices found through "Network Search" can execute network AT commands within 1 minute. Users can directly send AT commands to the device's IP and port. The timer will refresh after the device executes the command. It should be noted that the network AT mode does not affect the working mode of the serial port.

For detailed description of AT commands, please refer to the AT commands section.

### Firmware Upgrade

This series of devices supports two ways to upgrade the firmware. Generally, the upgrade does not affect the previous parameter configuration of the device.

#### HTTP Upgrade

The device supports HTTP protocol upgrade. Users use AT commands to trigger the upgrade action. The upgrade process is as follows:

1. Put the firmware in the directory of the HTTP server. Assume the server port is 8080 and the firmware URI is update.xxx.com:8080/firmware.bin .
2. Send AT command via serial or network: AT+DOWNLOAD =update.xxx.com :8080/ firmware.bin
3. If the download is successful, the device will respond with OK, otherwise it will respond with FAIL
4. After the download is successful, you need to restart manually to upgrade

#### Web page upgrade

Log in to the device's built-in web page and enter **the system** page to perform the upgrade operation



## Exception handling

The device supports the no data restart function. When no data is received from the network within a certain period of time, the device will automatically restart. This function is enabled by default and the time is 24 hours . The setting command is:

AT+ SOCKRTO=time ( 0≤time≤4320 , 0 means turn off this function, unit: min)

# Parameter settings

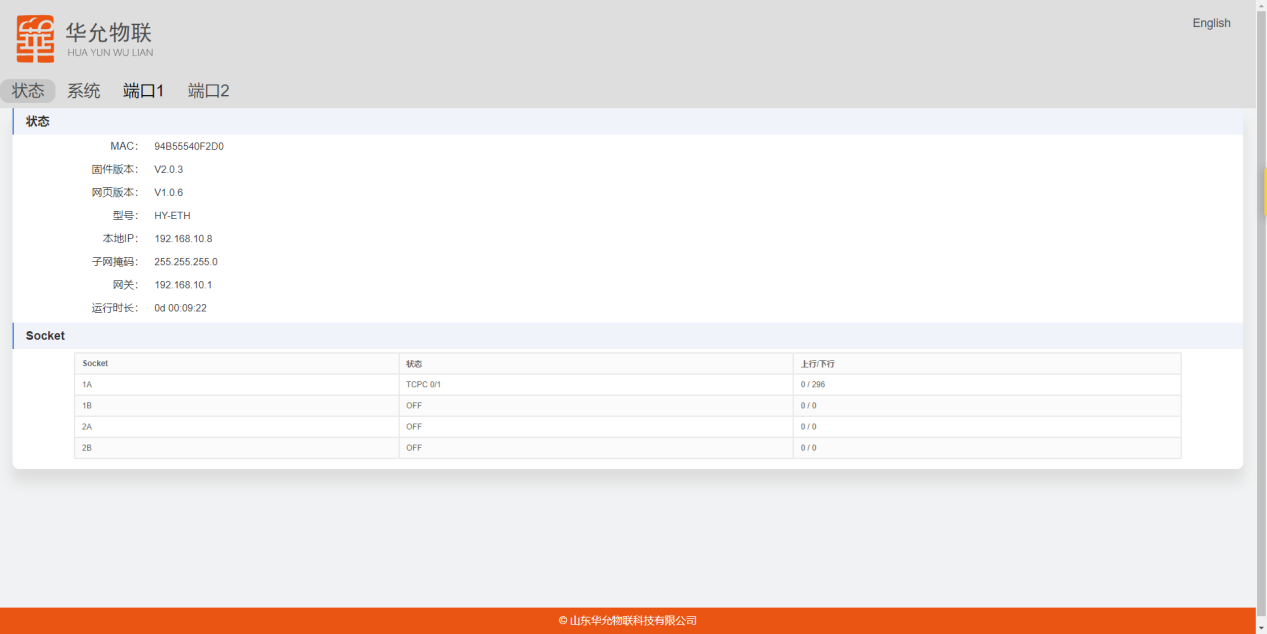
## Web page settings

The built-in web page allows customers to intuitively query device parameters, set up settings, and upgrade firmware.

Enter the device IP in the browser to open the device's built-in page. The default user name is admin and the password is admin.

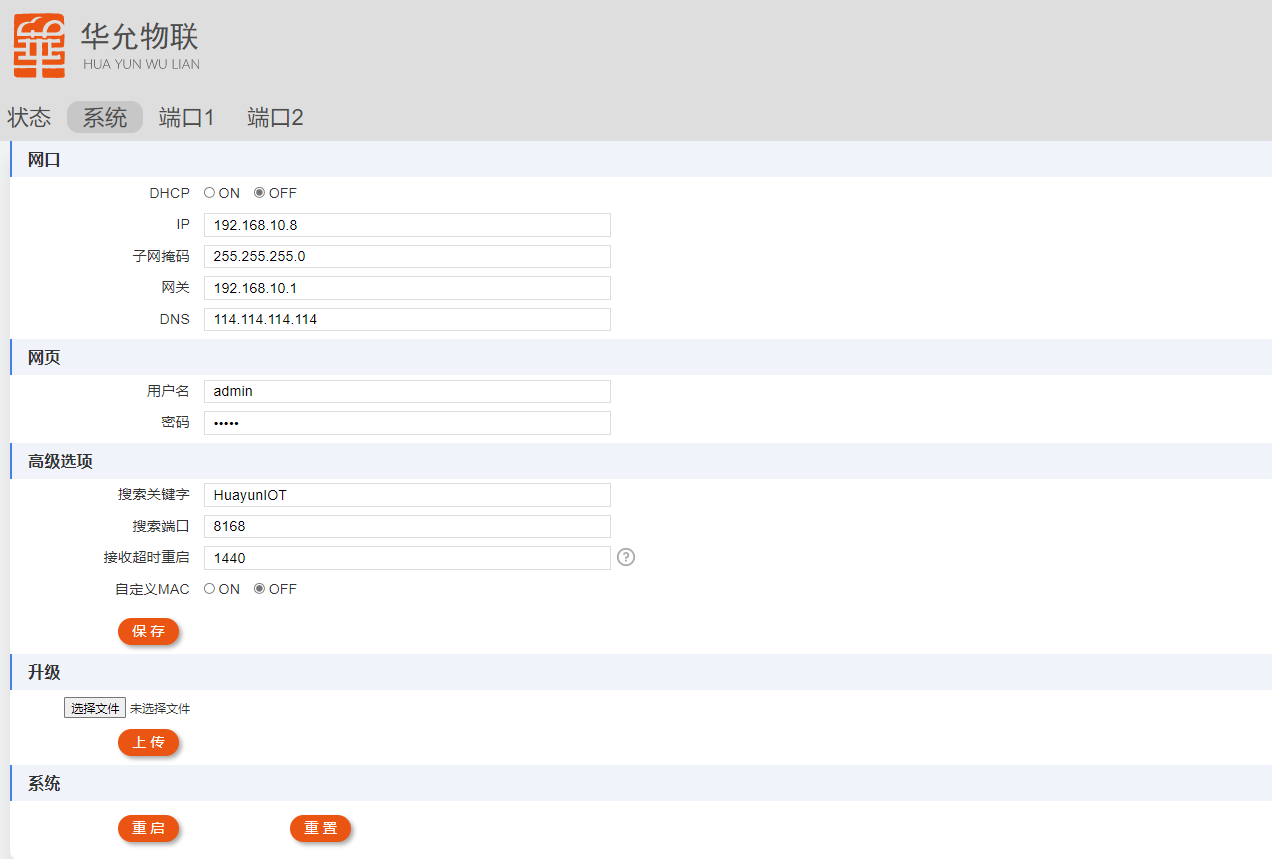
### Status Page

The status page contains basic device information.



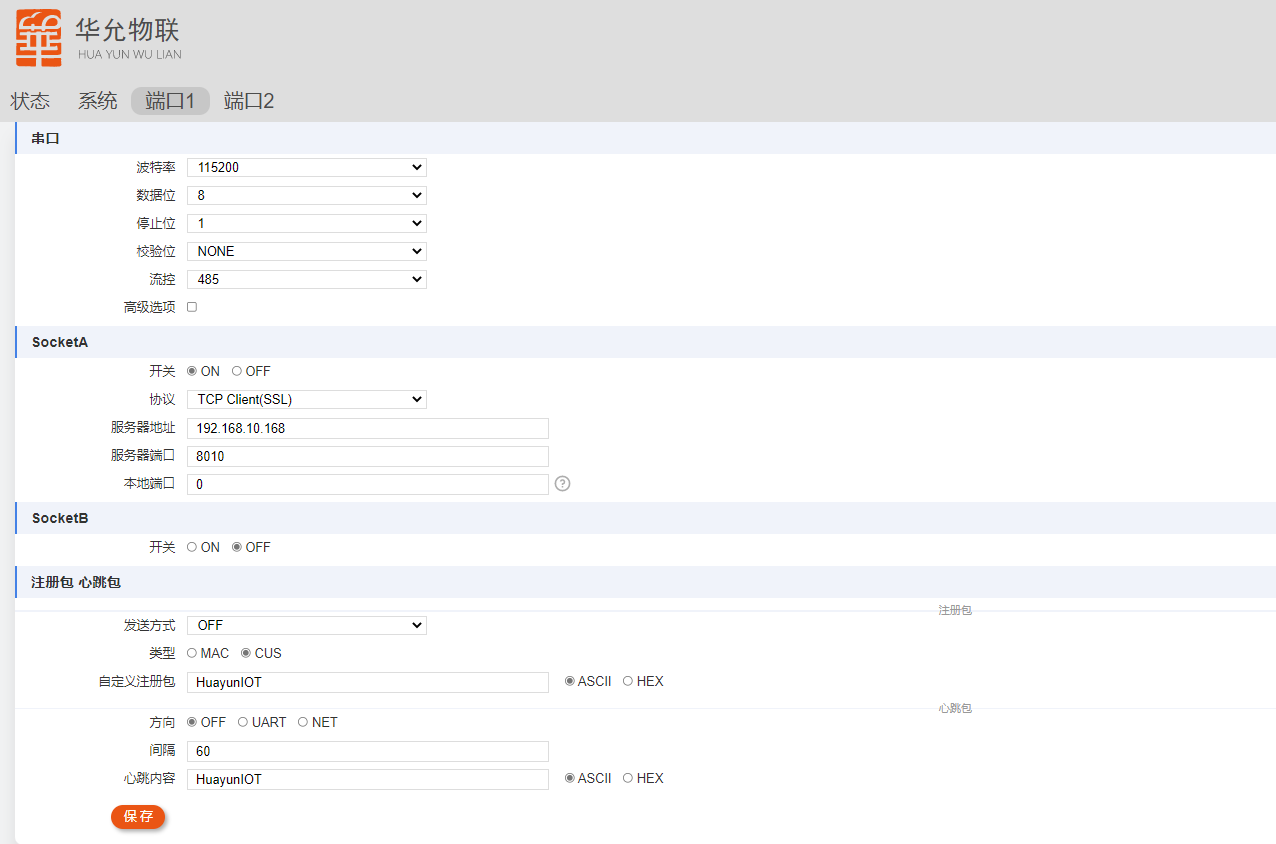
### System Management Page

The management page is related to device system management. You can configure the web username and password, network search port, keywords, etc. It also supports firmware upgrades.



### Port Settings Page

The port setting page is for each serial port and its corresponding network socket parameters, heartbeat packets, registration packets, etc.



## AT commands

AT commands can be used to query and set parameters. This product supports serial port AT and network AT. For serial port AT mode switching method, please refer to 2.1.3 Serial Port Working Mode , and for network AT entry method, please refer to 2.4.2 Network AT .

After setting the parameters, they are automatically saved and take effect after restart.

### AT command format

AT commands follow the following format:

1. Starts with AT+ and ends with \r\n
2. The command is not case sensitive, it is recommended to use uppercase letters
3. The command line can only contain one AT command, and the maximum length of a single command is 2 56 bytes.
4. are separated by commas .
5. You must wait for the previous command to return a result before sending a new command (the maximum command timeout is 5 seconds)

There are three types of commands: query, setting, and help. The format of each type is as follows:

* Query

Send: AT+ CMD\r\n or AT+CMD ?\r\n

Return: \ r\n + CMD:value\r\nOK\r\n

* set up

Send: A T+CMD=value1,value2…\r\n

Returns: \ r\nOK\r\n

* Help (only available for commands that can set parameters)

Send: A T+CMD=?\r\n

Return: \r\n+CMD:(parameter 1: range),(parameter 2: range) … \r\n OK\r\n

The following instructions are omitted unless otherwise specified .

### AT command error code information

When the command fails to execute, the device will issue an error code. The error code format is: \ r\n+ERROR:Error\_Code\r\nThe description of the error code is shown in Table 3 Error Code

Table 3 Error code

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Error Type** | **reason** |
| ARGS | Illegal parameters | The parameter length, size, format, etc. are illegal. |
| A RGC | Invalid number of parameters | Wrong number of parameters |
| CMD\_UNKNOWN | Unknown command | Instruction does not exist |
| CMD\_FORMAT | Format Error | start with AT + |
| CMD\_LENGTH | Wrong length | Exceeded maximum command length |
| DEV\_MEMORY | Memory Error | Memory Error |
| DEV\_SAVE | Save failed | Save failed |

### Quickly master AT commands

The large number of AT commands of the product brings flexibility to users, but also increases the difficulty for users to get started with AT commands. This section introduces several commonly used commands so that users can master most of the commands in a short time.

* Switch to transparent transmission mode: AT+EXIT
* Get command list: AT+LIST
* Reboot the device: AT+REBOOT
* Restore default parameters: AT+ RSTCFG
* Query parameter format: AT+CMD =? For example: send AT+ ECHO =? Return +ECHO: (echo: OFF/ON)

### Detailed explanation of A T instruction

This section introduces the AT commands of the device in detail. Note that when the command contains a lowercase letter n , it means that the command belongs to the nth serial port. Please replace n with the serial port number when sending the command .

For example, to query the parameters of serial port 1 : AT+UART1

there is a lowercase letter s in the command , it means SocketA or SocketB. Please replace s with A or B when sending the command . For example, to set the SocketA parameters of serial port 1: AT+SOCK1A=TCPC,192.168.1.8,8888

#### LIST displays a list of commands

|  |  |
| --- | --- |
| AT+LIST displays the command list | |
| Query  AT+LIST | +LIST: (command list)  OK |

#### EXIT Exit command mode

|  |  |
| --- | --- |
| AT+EXIT Exit command mode | |
| AT+EXIT | OK |

#### MAC query device MAC

|  |  |  |
| --- | --- | --- |
| AT+MAC query device MAC | | |
| AT+MAC | | +MAC:<mac>  OK |
| parameter | | |
| <mac> | MAC address, example: 0080E1134527 | |

#### VER query firmware version

|  |  |  |
| --- | --- | --- |
| AT+VER query firmware version | | |
| AT+VER | | +MAC:<ver>  OK |
| parameter | | |
| <ver> | Version, example: V1.0.0 | |

#### DEVINFO query device information

|  |  |  |
| --- | --- | --- |
| AT+ DEVINFO query firmware version | | |
| AT+DEVINFO | | +MODULE:<value>  +VERSION:<value>  +MAC:<value>  +DECRYPT:<value>  +BUILD:<value>  +PRODUCT TIME:<value>  +SN:<value> |
| parameter | | |
| <MODULE> | Product Model | |
| <VERSION> | Firmware version | |
| <MAC> | MAC | |
| <DECRYPT> | Firmware decryption status, undecrypted functions are limited | |
| <BUILD> | Compile time | |
| <PROTIME> | Production time | |
| <SN> | Product serial number | |

#### REBOOT Reboot the device

|  |  |
| --- | --- |
| AT+REBOOT restarts the device | |
| AT+REBOOT | OK |

#### RSTCFG restores backup parameters and automatically restarts

|  |  |
| --- | --- |
| AT+RSTCFG restores the backup parameters and automatically restarts (the effect is equivalent to the Reload button) | |
| AT+RSTCFG | OK |

#### BK CFG backup current operating parameters

|  |  |
| --- | --- |
| AT+BKCFG backup current running parameters | |
| AT+BKTCFG | OK |

#### CLRCFG restores factory-set parameters and automatically restarts

|  |  |
| --- | --- |
| AT+CLRCFG restores the factory-fixed parameters and automatically restarts | |
| AT+CLRCFG | OK |

The device has three parameter partitions, and the relationship between them is as follows:

* Parameter area: This area is used to read the operating parameters when the device is turned on; it is used by users to query and modify parameters.
* Backup area: used for backing up and restoring the "parameter area".
* Curing area: Manufacturer-cured parameters, used to completely restore factory parameters.

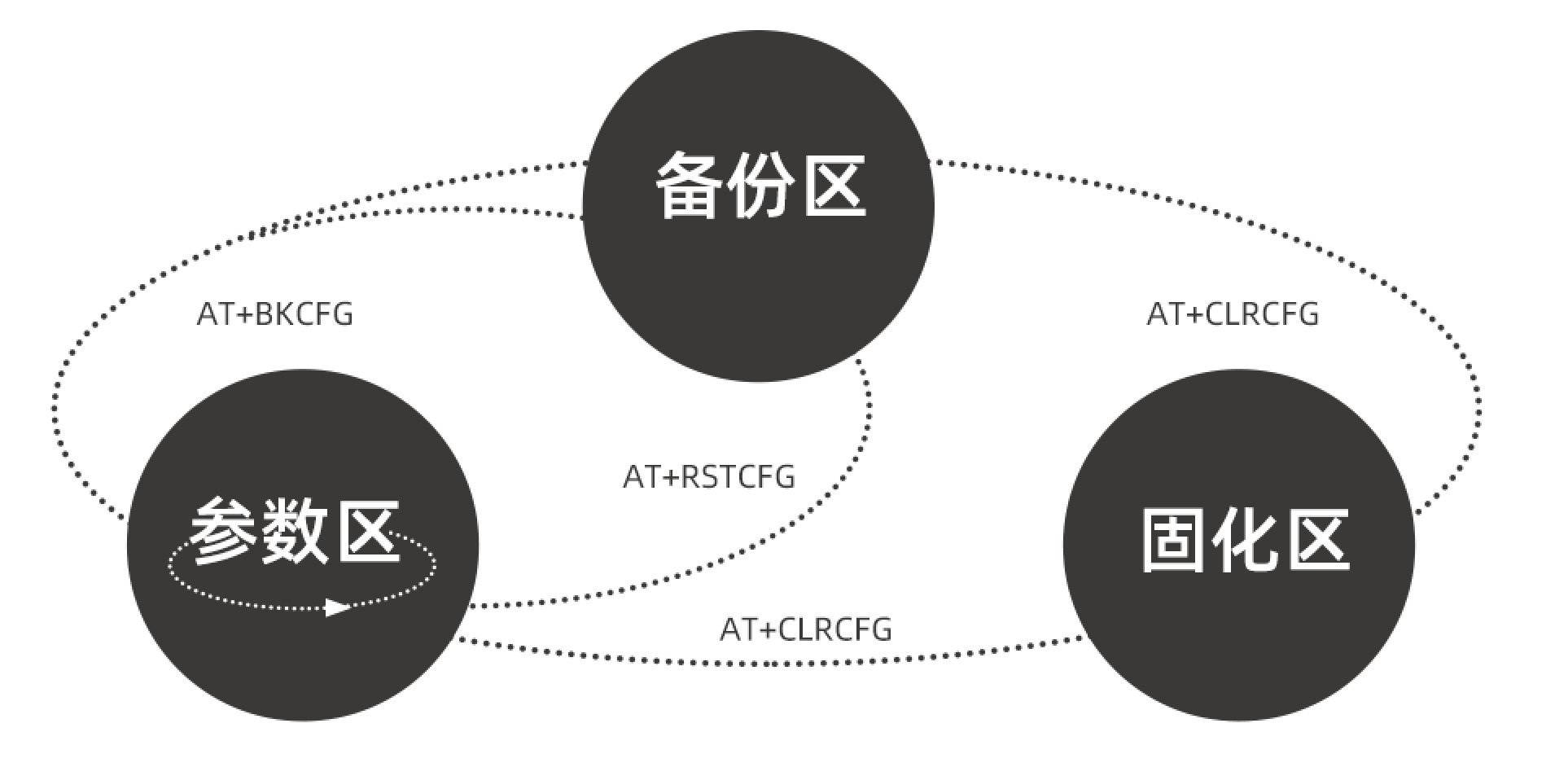


Figure 12 Parameter conversion

Recommended configuration process:

1. Users configure parameters according to their own needs
2. Save the parameters as default settings via BKCFG
3. Restore parameters via RSTCFG when necessary

#### SOCKLK queries TCP Client connection status

|  |  |  |
| --- | --- | --- |
| AT+SOCKLK queries the TCP Client connection status (UDP and TCP Server query results are OFF) | | |
| Query all socket connection status  AT+SOCKLK | | +SOCKLK:ns,<state>  +SOCKLK:ns,<state>  …  OK |
| Query the connection status of a TCP Client  n is the serial port number, s is A or B  AT+SOCKLK=ns | | +SOCKLK:<state>  OK |
| parameter | | |
| <state> | OFF: Disconnect  ON: Connected | |

#### AP LK STA information connected in AP mode

|  |  |  |
| --- | --- | --- |
| AT+APLK STA information connected in AP mode | | |
| AT+APLK | | +APLK:<n>  <mac>,<rssi>  …  OK |
| parameter | | |
| <n> | Number of STA accesses | |
| <mac> | MAC address of the STA | |
| <rssi> | RSSI signal strength | |

#### STA LK AP information connected in STA mode

|  |  |  |
| --- | --- | --- |
| AT+STALK Accessed AP information in STA mode | | |
| AT+STALK | | +APLK:<state>[,<ssid>,<mac>,<rssi>]  OK |
| parameter | | |
| <state> | Connection status, 1 means connected, 0 means not connected | |
| <ssid> | Name of the access AP | |
| <mac> | MAC address of the access AP | |
| <rssi> | RSSI signal strength | |

#### SCAN Scans surrounding APs

|  |  |  |
| --- | --- | --- |
| AT+STALK Accessed AP information in STA mode | | |
| AT+SCAN | | +SCAN:<n>  <rssi>,<ssid>,<auth>  …  OK |
| parameter | | |
| <n> | Number of APs scanned | |
| <ssid> | Name of the AP | |
| <mac> | AP's MAC address | |
| <auth> | Encryption | |

#### DOWNLOAD Network Upgrade

|  |  |  |
| --- | --- | --- |
| AT+DOWNLOAD HTTP protocol network upgrade command | | |
| AT+DOWNLOAD=uri | | <state> |
| parameter | | |
| <uri> | Firmware address. If the port is not specified, the default is 80. For example, the following address:  update.xxx.com/firmware.bin  192.168.1.56:8080/firmware.bin | |
| <state> | OK: Firmware downloaded successfully, manual restart is required to upgrade  +ERROR: FAIL: Download failed | |

#### PING Command

|  |  |  |
| --- | --- | --- |
| AT+PING | | |
| AT+PING=addr | | +PING:<result> |
| parameter | | |
| <addr> | IP or domain name | |
| <result> | Network not available: The network is not connected  Timeout: Timeout  Unknown host: unknown address  Number: ping delay, unit: ms | |

#### Name and password in AP mode

|  |  |  |
| --- | --- | --- |
| AT+AP Hotspot name and password in AP mode | | |
| AT+AP | | +AP:<ssid>,<pass> |
| AT+AP=<ssid>,<pass> | | OK |
| parameter | | |
| <ssid> | Hotspot name, 1~31 bytes | |
| <pass> | Wireless password, 8~63 bytes or NONE (no encryption) | |

#### W​ AP Channel

|  |  |  |
| --- | --- | --- |
| AT+WCH AP mode channel | | |
| AT+WCH | | +WCH:<ch> |
| AT+WCH=<ch> | | OK |
| parameter | | |
| <ch> | The channel of AP mode, the value range is 1~13, the default is 1 | |

#### NUMSTA The maximum number of STAs that can access AP mode

|  |  |  |
| --- | --- | --- |
| AT+NUMSTA The maximum number of STAs added in AP mode | | |
| AT+NUMSTA | | +NUMSTA:<n> |
| AT+NUMSTA=<n> | | OK |
| parameter | | |
| <n> | The maximum number of connected STAs, the value range is 1~6, the default is 3 | |

#### IP settings in LAN AP mode

|  |  |  |
| --- | --- | --- |
| AT+LAN AP mode IP | | |
| AT+LAN | | +LAN:<ip>,<mask> |
| AT+AP=<ip>,<mask> | | OK |
| parameter | | |
| <ip> | LAN port IP, default is 192.168.4.1 | |
| <mask> | LAN port mask, the default is 255.255.255.0 | |

#### Information about STA accessing hotspots

|  |  |  |
| --- | --- | --- |
| AT+ Information about STA accessing hotspots | | |
| AT+STA | | +STA:<ssid>,<pass> |
| AT+STA=<ssid>,<pass> | | OK |
| parameter | | |
| <ssid> | Hotspot name, 1~31 bytes | |
| <pass> | Password, 8~63 bytes or NONE (no encryption) | |

#### ECHO command echo switch

|  |  |  |
| --- | --- | --- |
| AT+ECHO command echo switch | | |
| AT+ECHO | | +ECHO:<state> |
| AT+ECHO=<state> | | OK |
| parameter | | |
| <state> | ON: Enable  OFF: Disable (default value) | |

#### BOOTINFO boot information

|  |  |  |
| --- | --- | --- |
| AT+BOOTINFO startup information | | |
| AT+BOOTINFO | | +BOOTINFO:<info> |
| AT+BOOTINFO=<info> | | OK |
| parameter | | |
| <info> | 1~16 byte string, default value: Start | |

#### SOCKRTO network no data reception timeout restart interval

|  |  |  |
| --- | --- | --- |
| AT+SOCKRTO Network no data reception timeout restart interval | | |
| AT+SOCKRTO | | +SOCKRTO:<time> |
| AT+SOCKRTO=<time> | | OK |
| parameter | | |
| <time> | Value range: 0~65535, default 1440 (24 hours) 0 means turn off this function  Unit: min | |

#### WEBU Username and Password

|  |  |  |
| --- | --- | --- |
| AT+WEBU web page username and password | | |
| AT+WEBU | | +WEBU:<user>,<pwd> |
| AT+WEBU=<user>,<pwd> | | OK |
| parameter | | |
| <user> | User name, 1~16 bytes string, default value: admin | |
| <pwd> | Password, 1~16 bytes string, default value: admin | |

#### LANG Webpage language

|  |  |  |
| --- | --- | --- |
| AT+LANG web page language | | |
| AT+LANG | | +LANG:<lang> |
| AT+LANG=<lang> | | OK |
| parameter | | |
| <lang> | CN: Chinese (default)  EN：English | |

#### WAN port parameters

|  |  |  |
| --- | --- | --- |
| AT+WAN port parameters | | |
| AT+WAN | | +WAN:<dhcp\_en>,<ip>,<gw>,<mask> |
| AT+WAN=<dhcp\_en>,<ip>,<gw>,<mask> | | OK |
| parameter | | |
| <dhcp\_en> | DHCP: Dynamic IP. The following parameters can be omitted when setting a dynamic IP.  STATIC: Static IP (default) | |
| <ip> | IP under static IP (default 192.168.10.8) | |
| <gw> | Gateway under static IP (default 192.168.10.1) | |
| <mask> | Subnet mask under static IP (default 255.255.255.0) | |

#### DNS backup DNS address

|  |  |  |
| --- | --- | --- |
| AT+DNS alternate DNS address | | |
| AT+DNS | | +DNS:<addr> |
| AT+DNS=<addr> | | OK |
| parameter | | |
| <addr> | Alternate DNS address, default is 114.114.114.114  (The preferred DNS is the gateway address) | |

#### SEARCH Network search port and keyword

|  |  |  |
| --- | --- | --- |
| AT+SEARCH backup SEARCH address | | |
| AT+SEARCH | | +SEARCH:<port>,<key> |
| AT+SEARCH=<port>,<key> | | OK |
| parameter | | |
| <port> | Port, value range 1~65535, default 8168 | |
| <key> | Search keyword, 1~16 bytes | |

#### UART n serial port parameters

|  |  |  |
| --- | --- | --- |
| AT+UARTn query and set the parameters of the nth serial port | | |
| AT+UARTn | | +UARTn:<baudrate>,<databits>,<stopbits>,<parity>,<fc> |
| AT+UARTn=<baudrate>,<databits>,<stopbits>,<parity>,<fc> | | OK |
| parameter | | |
| <baudrate> | Baud rate, default value is 115200  Optional 2400/4800/9600/19200/38400/57600/115200/230400/460800 | |
| <databits> | Data bits, 7 or 8 (default) Note: E31xx series does not support 7 data bits | |
| <stopbits> | Stop bits, 1 (default) or 2 | |
| <parity> | Check digit: NONE (default value)/EVEN/ODD | |
| <fc> | Flow control, NFC (default)/485 | |

#### UARTTL n serial port packet interval and length

|  |  |  |
| --- | --- | --- |
| AT+UARTTLn serial port n packet interval and length | | |
| AT+UARTTLn | | +UARTTLn:<tm>,<len> |
| AT+UARTTLn=<tm>,<len> | | OK |
| parameter | | |
| <tm> | Packing interval, value range 1~300ms, default 5 | |
| <len> | Packing length, value range 8~1460, default 1024 | |

#### SOCKE Nn s serial port n corresponding to the SocketA / B switch

|  |  |  |
| --- | --- | --- |
| AT+SOCKENns Socket A/B switch corresponding to serial port n (n is the serial port number, s is A or B) | | |
| AT+SOCKENns | | +SOCKENns:<state> |
| AT+SOCKENns:<state> | | OK |
| parameter | | |
| <state> | ON: Enabled. By default, only Socket A of serial port 1 is enabled.  OFF: Off | |

#### SOCK n s Socket parameters​

|  |  |  |
| --- | --- | --- |
| AT+SOCKns Socket parameters (n is the serial port number, s is A or B) | | |
| AT+SOCKns | | +SOCKns:<type>,<addr>,<port>[,localport] |
| AT+SOCKns:<type>,<addr>,<port>[,localport] | | OK |
| parameter | | |
| <type> | Protocol Type:  TCPC: TCP Client  TCPS: TCP Server  UDPC: UDP Client  UDPS: UDP Server  HTPC: HTTP Client | |
| <addr> | Remote server address, within 64 bytes, invalid when used as a server | |
| <port> | In Client mode, it is the remote server port; in Server mode, it is the local port  Value range: 1~65535 | |
| <localport> | Local port. This parameter can be omitted.  This option is only valid under TCPC or UDPC. It is recommended to use a random port under TCPC and a fixed local port under UDPC.  Note: The same end cannot be reused for the same protocol.  Value range: 0~65535, 0 means using a random port (default value: 0) | |

#### HTPURL n HTTP URL Parameters

|  |  |  |
| --- | --- | --- |
| AT+HTPURLn HTTP URL parameters | | |
| AT+HTPURLn | | +HTPURLn:<url> |
| AT+HTPURLn:<url> | | OK |
| parameter | | |
| <url> | 1~64 bytes string, URL parameter is only valid in POST request mode | |

#### HTTP Header Parameters​​

|  |  |  |
| --- | --- | --- |
| AT+HTPHDn HTTP Header Parameters | | |
| AT+HTPHDn | | +HTPHDn:<hd> |
| AT+HTPHDn:<hd> | | OK |
| parameter | | |
| <hd> | HTTP protocol header, multiple headers are separated by vertical lines | , length range 1~64 bytes | |

#### HTPFT n Whether to filter the packet header of the data received by HTTP

|  |  |  |
| --- | --- | --- |
| AT+HTPFTn Whether to filter the packet header of HTTP data received by serial port n | | |
| AT+HTPFTn | | +HTPFTn:<state> |
| AT+HTPFTn:<state> | | OK |
| parameter | | |
| <state> | ON: Enable filtering (default)  OFF: Disable filtering | |

#### HTPREQ n HTTP request method

|  |  |  |
| --- | --- | --- |
| AT+HTPREQn HTTP request method | | |
| AT+HTPREQn | | +HTPREQn:<request> |
| AT+HTPREQn:<request> | | OK |
| parameter | | |
| <request> | GET: GET request  POST: POST request  AUTO: Automatic mode, automatically distinguish the request method according to the data when sending a request | |

#### HTTP request timeout

|  |  |  |
| --- | --- | --- |
| AT+HTPTOn HTTP request timeout | | |
| AT+HTPTOn | | +HTPTOn:<to> |
| AT+HTPTOn:<to> | | OK |
| parameter | | |
| <to> | The value range is 1~30s, the default is 6 | |

#### MQCONF n MQTT connection parameter configuration

|  |  |  |
| --- | --- | --- |
| AT+MQCONFn (n is the serial port number) | | |
| AT+MQCONFn | | +MQCONFn:<ver>,<clean>,<keepalive> |
| AT+MQCONFn =<ver>,<clean>,<keepalive> | | OK |
| parameter | | |
| <ver> | MQTT version  3:3.1  4: 3.1.1 (default) | |
| <clean> | Is the connection session clear?  0: No  1: Yes (default) | |
| <keepalive> | Keepalive heartbeat interval, range 30~65535 seconds | |

#### MQAUTH n MQTT Authentication

|  |  |  |
| --- | --- | --- |
| AT+MQAUTHn (n is the serial port number) | | |
| AT+MQAUTHn | | +MQAUTHn:<id>,<user>,<pass> |
| AT+MQAUTHn =<id>,<user>,<pass> | | OK |
| parameter | | |
| <id> | Client ID, the device ID connected to the same client cannot be repeated, 1~64 bytes | |
| <user> | User name, 1~64 bytes | |
| <pass> | Password, 1~128 bytes | |

#### MQSUB n MQTT Subscription Settings

|  |  |  |
| --- | --- | --- |
| AT+MQSUBn (n is the serial port number) | | |
| AT+MQSUBn | | +MQSUBn:<enable>,<topic>,<qos> |
| AT+MQSUBn =<enable>,<topic>,<qos> | | OK |
| parameter | | |
| <enable> | Whether to enable subscription:  1: Enable (default)  0: Off | |
| <topic> | Subscribed topic, 1~64 bytes | |
| <qos> | QoS level:  0: QoS0 is sent at most once  1: QoS1 sent at least once  2: QoS2 ensures only once | |

#### MQPUB n MQTT publishing settings

|  |  |  |
| --- | --- | --- |
| AT+MQPUBn (n is the serial port number) | | |
| AT+MQPUBn | | +MQPUBn:<enable>,<topic>,<qos>,<retain> |
| AT+MQPUBn =<enable>,<topic>,<qos>,<retain> | | OK |
| parameter | | |
| <enable> | Whether to enable publishing:  1: Enable (default)  0: Off | |
| <topic> | Topic for publishing, 1~64 bytes | |
| <qos> | QoS level:  0: QoS0 is sent at most once (default)  1: QoS1 sent at least once  2: QoS2 ensures only once | |
| <retain> | Keep sending messages  0: No (default)  1: Yes | |

#### MQWILL n MQTT Last Wish Message

|  |  |  |
| --- | --- | --- |
| AT+MQWILLn (n is the serial port number) | | |
| AT+MQWILLn | | +MQWILLn:<enable>,<topic>,<qos>,<msg>,<retain> |
| AT+MQWILLn =<enable>,<topic>,<qos>,<msg>,<retain> | | OK |
| parameter | | |
| <enable> | Whether to enable:  1: On  0: Disable (default) | |
| <topic> | Topic, 1 to 64 bytes | |
| <qos> | QoS level:  0: QoS0 is sent at most once (default)  1: QoS1 sent at least once  2: QoS2 ensures only once | |
| <msg> | Message content, 1~64 bytes | |
| <retain> | Keep sending messages  0: No (default)  1: Yes | |

#### MQMD​ MQTT mode settings

|  |  |  |
| --- | --- | --- |
| AT+MQMDn (n is the serial port number) | | |
| AT+MQMDn | | +MQMDn:<mode> |
| AT+MQMDn =<mode> | | OK |
| parameter | | |
| <mode> | STD: Standard MQTT (default)  ALI: Alibaba Cloud mode, which can automatically calculate Alibaba Cloud's authentication information | |

#### REG TPn registration packet type

|  |  |  |
| --- | --- | --- |
| AT+REGTPn Registration packet type | | |
| AT+REGTPn | | +REGTPn:<type> |
| AT+REGTPn:<type> | | OK |
| parameter | | |
| <type> | MAC: Use MAC address as the registration packet, the format is 6-byte HEX array  CUS: Custom Registration Package | |

#### REG MDn registration packet sending mode

|  |  |  |
| --- | --- | --- |
| AT+REGMDn Registration packet type | | |
| AT+REGMDn | | +REGMDn:<mode> |
| AT+REGMDn:<mode> | | OK |
| parameter | | |
| <mode> | OFF: Disable the registration package function (default)  FIRST: TCP Client connection send or UDP Client first network send  EVERY: Data Carrying  ALL：FIRST+EVERY | |

#### REG DATn Customized Registration Package Contents

|  |  |  |
| --- | --- | --- |
| AT+REGDATn Customize registration package content | | |
| AT+REGDATn | | +REGDATn:<data> |
| AT+REGDATn:<data> | | OK |
| parameter | | |
| <data> | User-defined registration package, HEX string format, 2~64 bytes | |

#### HEART MDn heartbeat mode

|  |  |  |
| --- | --- | --- |
| AT+HEARTMDn heartbeat mode | | |
| AT+HEARTMDn | | +HEARTMDn:<mode> |
| AT+HEARTMDn=<mode> | | OK |
| parameter | | |
| <mode> | OFF: Disable (default)  UART: Serial heartbeat  NET: Network Heartbeat | |

#### HEARTTM nHeartbeat interval

|  |  |  |
| --- | --- | --- |
| AT+HEARTTMn heartbeat interval | | |
| AT+HEARTTMn | | +HEARTTMn:<time> |
| AT+HEARTTMn:<time> | | OK |
| parameter | | |
| <time> | The value range is 1~86400s, the default is 60 | |

#### HEARTDAT n Heartbeat packet content

|  |  |  |
| --- | --- | --- |
| AT+HEARTDATn Heartbeat content | | |
| AT+HEARTDATn | | +HEARTDATn:<data> |
| AT+HEARTDATn:<data> | | OK |
| parameter | | |
| <data> | User-defined registration package, HEX string format, 2~64 bytes | |

# Disclaimer

Shandong Huayun IoT Technology Co., Ltd. provides this document to support its customers' product design. Customers must design their products in accordance with the specifications and parameters provided in the document. The company does not assume any responsibility for personal injury or property damage caused by improper operation of customers. The company will not notify you of any changes to product specifications and product descriptions without prior notice.

# Contact Details

Company: Shandong Huayun IoT Technology Co., Ltd.

Address: 12-1, Building 4, Future Entrepreneurship Plaza, North Section of Gangxing 3rd Road, High-tech Zone, Jinan City, Shandong Province

Official website: http://www.huayuniot.com