VT-SOM-AH
Wi-Fi HaLow Module

User Manual

Version: 1.5

Revision History:

No.	Version	Description	Date
1	V1.0	First release	Dec. 31, 2025
2	V1.1	1. Added AT commands. 2. Added illustrative examples for the iPerf test 3. Added instructions for firmware upgrade	Jan. 13, 2026
3	V1.2	1. Modified the description of the 'AT+GPIOCONF' command. 2. Modified the use of the 'AT+WIPERF' command. 3. Added AT commands: 'AT+STCPKEEPALIVE', 'AT+STCPNODELAY'.	Jan. 20, 2026
4	V1.3	Modified the use of the 'AT+GPIOVAL' command.	Jan. 30, 2026
5	V1.4	Modified examples for the 'AT+STCPKEEPALIVE' & 'AT+STCPKEEPALIVE' commands.	Feb. 5, 2026
6	V1.5	1. Included VT-SOM-AH (6108) in the manual and updated related description accordingly. 2. Updated Chapter 1 to add description of DTU Transmission Mode for data transfer and removed Chapter 2 on driver download instructions based on software upgrade. 3. Updated AT+VER, AT+SRECVINFO commands. 4. Added AT+RECV, AT+RECVMODE, AT+SOCKET, AT+WSSID, +++++ commands. 5. Updated Chapters 3-5 to include DTU Transmission Mode. 6. Added Chapter 6 to document the procedures for firmware flashing.	Apr. 23, 2026

Table of Contents

Foreword

This manual is intended to provide instructions on the setup, networking, and operation of Vantron's Wi-Fi HaLow system-on-modules (SOMs).

Currently available Wi-Fi HaLow SOM models include: VT-SOM-AH-8108, VT-SOM-AH-8108-M2, and VT-SOM-AH (6108).

As the modules support both USB and UART interfaces to facilitate user development, you can select the module packaging according to actual application requirements.

Intended Users

This manual is intended for:

- Embedded system engineers
- Developers
- Technical support engineers
- Other users

Copyright

Disclaimer

While all information contained herein has been carefully checked to assure its accuracy in technical details and typography, Vantron does not assume any responsibility resulting from any error or features of this manual, nor from improper uses of this manual or the software.

It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the product may be changed without notice.

Technical Support and Assistance

Should you have any question about the EVB that is not covered in this manual, contact your sales representative for solution. Please contain the following information in your question:

- Product name and PO number;
- Complete description of the problem;
- Error message you received, if any.

Vantron Technology, Inc.

Address: 48434 Milmont Drive, Fremont, CA 94538

Tel: (650) 422-3128

Email: sales@vantrontech.com

Symbology

This manual uses the following signs to prompt users to pay special attention to relevant information.

⚠	Caution for latent damage to system or harm to personnel
i	Attention to important information or regulations

General Safety Instructions

The module is supposed be installed by knowledgeable, skilled persons familiar with local and/or international electrical codes and regulations. For your safety and prevention of damage to the module and other equipment connected to it, please read and observe carefully the following safety instructions prior to installation and operation. Keep this manual well for future reference.

- Do not disassemble or otherwise modify the module. Such action may cause heat generation, ignition, electronic shock, or other damages including human injury, and may void your warranty.
- Keep the module away from heat source, such as heater, heat dissipater, or engine casing.
- Avoid foreign materials to stain the module as they may cause the module to malfunction or burn out.
- To ensure proper functioning and prevent overheating of the module, do not cover the module with non-ventilated materials.
- Follow the installation instructions with the installation tools provided or recommended.
- The use or placement of the operation tools shall comply with the code of practice of such tools to avoid short circuit of the module.
- Use only approved antenna(s). Non-approved antenna(s) may produce spurious or excessive RF transmitting power which may violate FCC limits.
- Power off and contact Vantron technical support engineer in case of the following faults:
    - The module is damaged.
    - The temperature is excessively high.
    - Fault is still not solved after troubleshooting according to this manual.
- Do not use in combustible and explosive environment:
    - Keep away from combustible and explosive environment.
    - Keep away from all energized circuits.

CHAPTER 1 INTRODUCTION

1.1 Product Overview

Vantron's Wi-Fi HaLow system-on-modules (SOMs) deliver a complete Wi-Fi HaLow connectivity solution. They integrate the Morse Micro MM6108/8108 single-chip SoC, which is compliant with the IEEE 802.11ah standard, along with an Arm® Cortex®-M33 ultra-low-power MCU.

These modules are designed to be readily integrated into any embedded device to provide a simplified Wi-Fi HaLow connection for customers looking to easily upgrade their prior RF technology to a Wi-Fi HaLow connection while using the latest WPA3 security protocol.

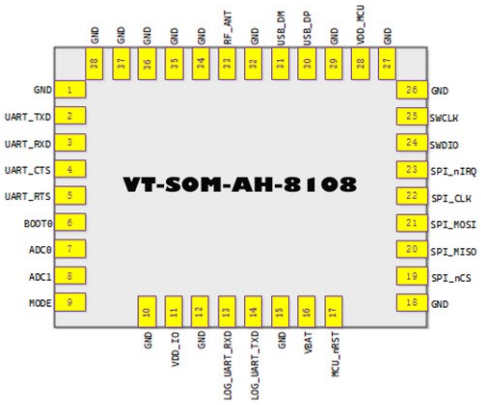
Operating in the 850~950MHz band with configurable channel widths of 1, 2, 4, and 8MHz, the modules support a variety of peripheral interfaces, including I^2C, UART, USB, CAN, ADC, and GPIO. They support the Station (STA) role in the MAC layer, making them ideal for connecting low-power NFC tags, sensors, and access control systems.

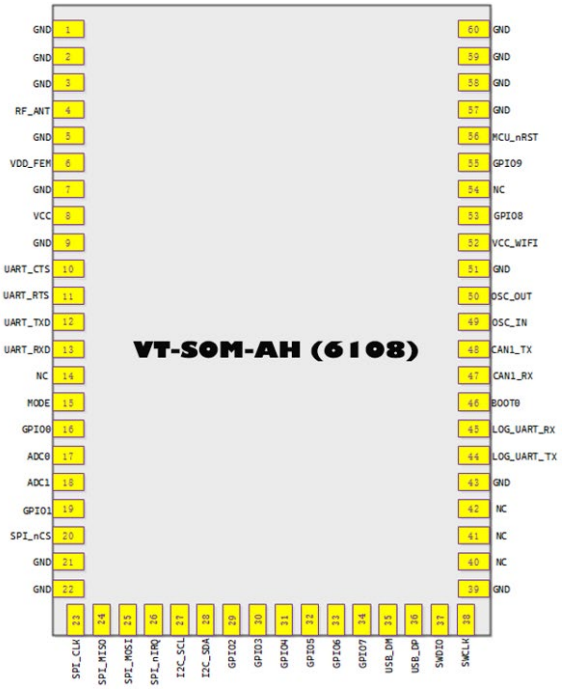
Currently available Wi-Fi HaLow SOM models include: VT-SOM-AH-8108, VT-SOM-AH-8108-M2, and VT-SOM-AH (6108).

The VT-SOM-AH-8108 and VT-SOM-AH-8108-M2 are powered by the MM8108 HaLow chipset, and the latter is packaged in the M.2 (2230) form factor.

The VT-SOM-AH (6108) is powered by the MM6108 HaLow chipset, offering a cost-effective solution.

1.2 Pinout





BOTTOM SIDE	
74	3.3V
72	3.3V
70	
68	
66	
64	
62	
60	
58	
56	#Disable
54	RESET_N
52	
50	
48	
46	
44	
42	
40	
38	
36	UART_CTS
34	UART_RTS
32	UART_RX
30	Key E
28	Key E
26	Key E
24	Key E
22	UART_TX
20	
18	GND
16	
14	
12	
10	
8	
6	
4	3.3V
2	3.3V

TOP SIDE	
75	GND
73	
71	
69	GND
67	
65	
63	GND
61	
59	
57	GND
55	
53	
51	GND
49	
47	
45	GND
43	
41	
39	GND
37	
35	
33	GND
31	Key E
29	Key E
27	Key E
25	Key E
23	
21	
19	
17	
15	
13	
11	
9	
7	GND
5	USB_DN
3	USB_DP
1	GND

VT-SOM-AH-8108-M2

I/O Description:

Name	Signal	Description
UART	UART_TXD	UART interface with flow control for module communication
	UART_RXD	
	UART_CTS	
	UART_RTS	
USB	USB_DP	USB interface for module communication
	USB_DM	
ADC	ADC0	12-bit ADC
	ADC1	
GPIO	GPIOx	General-purpose I/O
Console (Debug)	LOG_UART_TXD	Debug UART for system log output
	LOG_UART_RXD	
SPI	SPI_nCS	Reserved SPI
	SPI_CLK	
	SPI_MISO	
	SPI_MOSI	
	SPI_nIRQ	
I^2C	I2C_SDA	Reserved I^2C
	I2C_SCL	
SWD	SWCLK	Firmware download port
	SWDIO	
RST	MCU_RST	Hardware reset for the module

1.3 Working Mode

To accommodate different application scenarios, Vantron's Wi-Fi HaLow SOMs support two operating modes:

(1) AT Command Mode

(2) DTU Transmission Mode

Mode	Best For	Key Feature
AT Command Mode	Systems with an MCU/CPU that can send commands.	**Host-controlled** – module waits for instructions.
DTU Transmission Mode	Systems without a controller (sensors, simple devices)	**Standalone operation** – module works automatically after configuration.

1.3.1 AT Command Mode

In this mode, the SOM only executes corresponding actions after receiving AT commands. This includes device networking, data transmission, socket creation, etc. — all are performed passively.

This mode is primarily used in the following scenario:

1.3.2 DTU Transmission Mode

Parameters for operation in DTU Transmission Mode must be configured in AT Command Mode first. After parameter configuration and module restart, the SOM operates automatically, such as connecting to the AP, creating sockets, and performing reconnection upon disconnection. When data is received via UART or USB, it is transmitted directly to the target device over the network.

This mode is primarily used in application scenarios where the user side does not include a controller (MCU/CPU):

1.3.3 Mode Switching

Mode switching between can be performed by sending the string "+++++" to the SOM via the UART or USB interface.

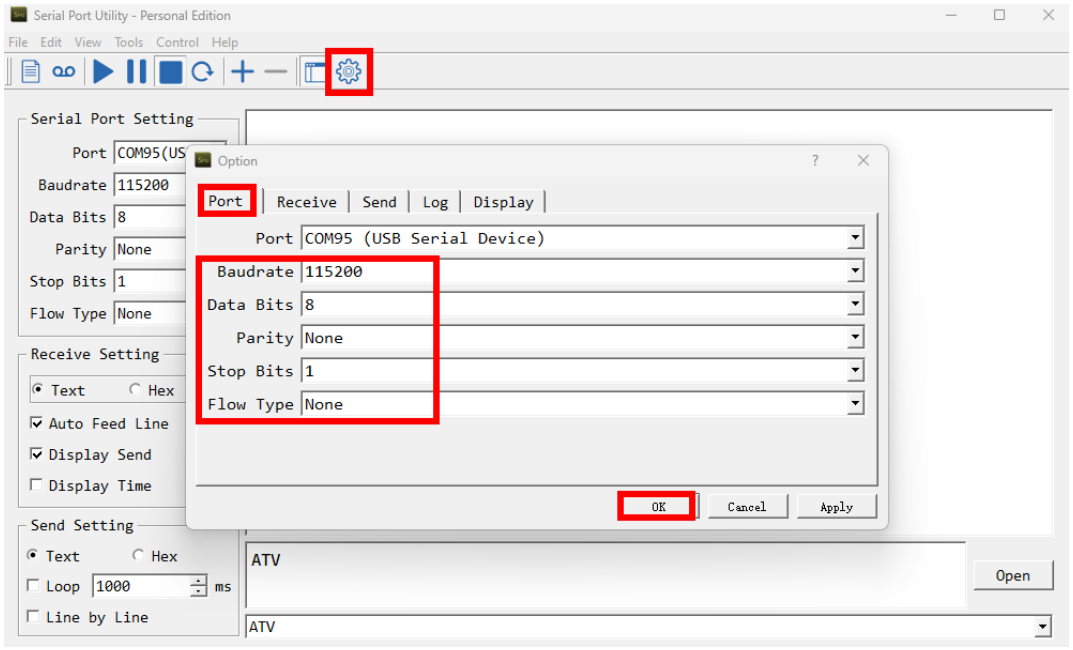
```
+++++
+BEVENT:"DTU_MODE"

+++++

+BEVENT:"AT_MODE"
```

When switching to the DTU Transmission Mode, remember to configure parameters in AT Command Mode before sending the string. After the module restarts, it enters the DTU Transmission Mode.

1.4 Serial Port Utility

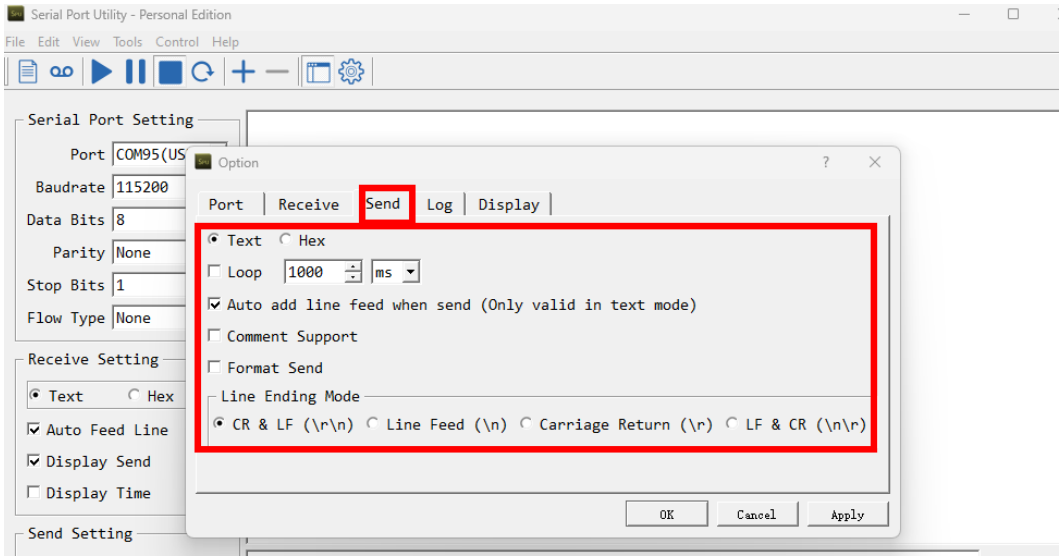
To quickly verify the module functionalities, a serial terminal is required. **Serial Port Utility** is recommended for sending AT commands on a **Windows** PC.

1. Set up the utility.
    - Port configuration

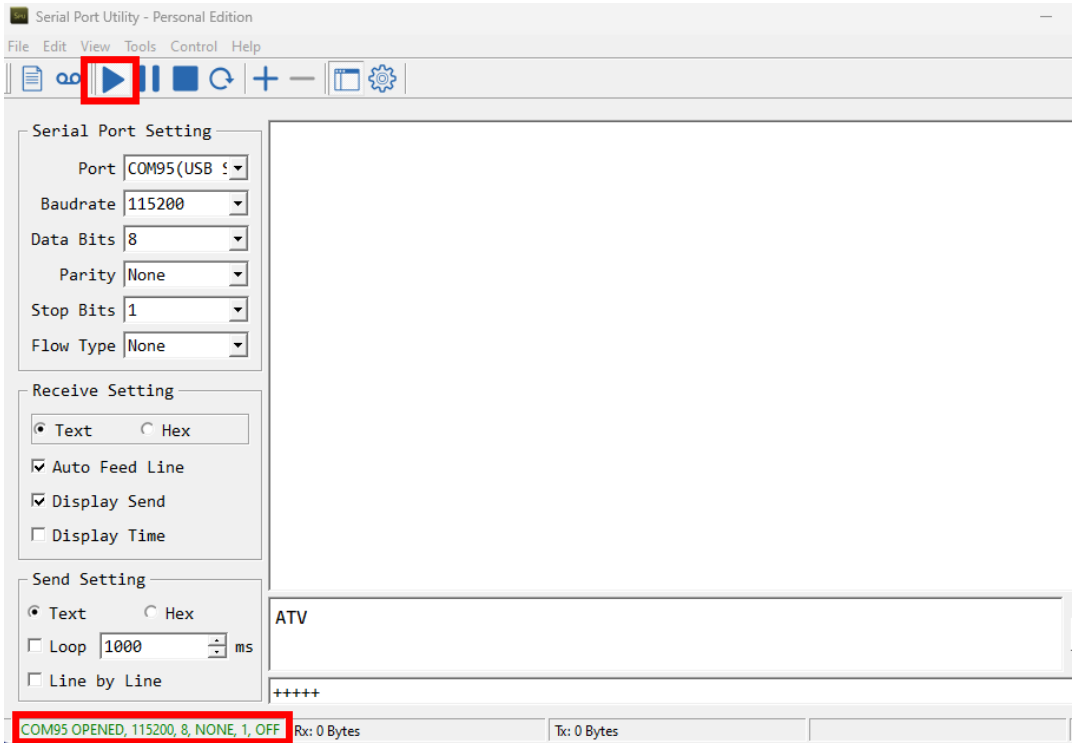


The port number shall correspond to the actual COMx number found under **Device Manager > Ports (COM & LPT)**.

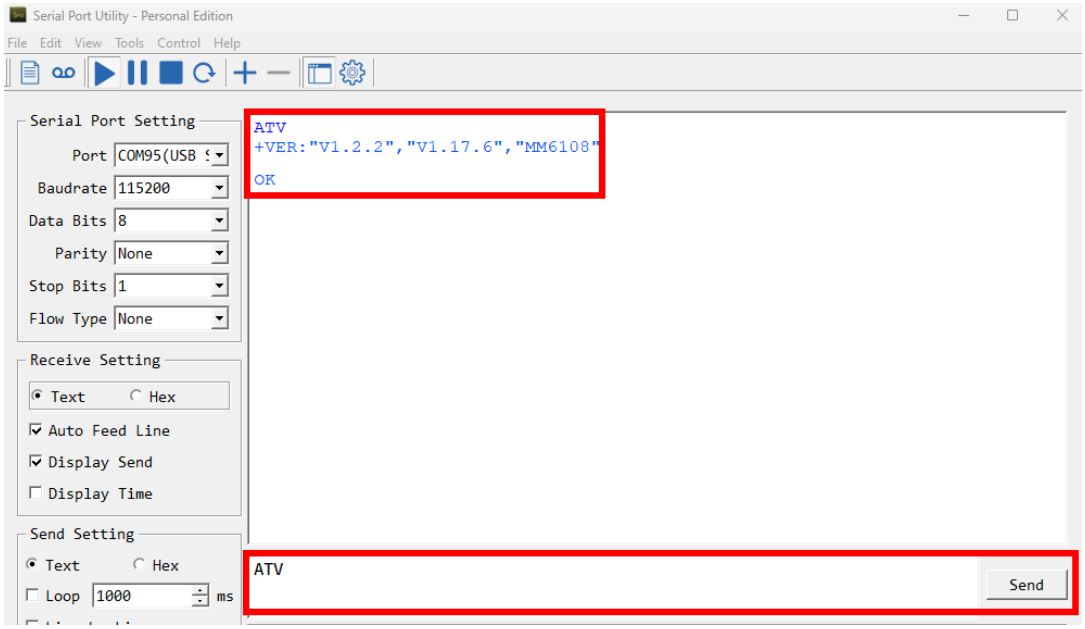
- Command sending parameters:



2. After configuration, click the **Connect** button. The message “COMx OPENED” indicates a successful connection between the module and the PC.



3. Enter commands in the send window and click **Send** to transmit.



CHAPTER 2 AT COMMANDS

Vantron's Wi-Fi HaLow SOMs come with a dedicated AT command set. Device configuration is implemented by sending AT commands through UART or USB.

All commands are sent in **AT Command Mode**. DTU-specific commands only **preconfigure** the parameters, which are automatically applied when switching to DTU Transmission Mode.

You are recommended to use **Serial Port Utility** to execute the commands. Remember to use '\r\n' as the line break after each command.

2.1 Command Structure

Type	Structure	Description
HELP	AT+<CMD>=?	List the format and description of input parameters.
SET / RUN	AT+<CMD> or AT+<CMD>=<x1,x2,…>	Execute a command with no parameters, or Set or run a parameterized command.
GET	AT+<CMD>? or AT+<CMD>?=<x1,x2,…>	Query the current value of a command, or Query the value of a command with specified parameters.

- String parameters **must** be enclosed in double quotes " ".
- < > denotes a **mandatory** field.
- [ ] denotes an **optional** field.
- All commands must terminate with a carriage return and line feed (CR-LF).

2.2 Command Responses

Response Message	Description
OK	Command executed successfully.
ERROR	Command not supported.
+<CMD>:1 ERROR	Invalid command parameters.
+<CMD>:2 ERROR	Command is still in progress.
+<CMD>:3 ERROR	Error occurred during execution or conditions not met.

2.3 General AT Commands

Command	Description
AT	Query the AT serial port (UART/USB) status.
ATE	Enable or disable command echo.
ATZ	Restore factory settings and reboot firmware.
AT+VER	Query firmware and software package versions.
AT+RST	Reset and reboot the module.
AT+BOOT	Get the system reboot cause.
AT+UART	Configure UART parameters.
AT+GPIOCONF	Configure GPIO pin mode, direction, and pull-up/down selection (**applicable to VT-SOM-AH (6108) only**).
AT+GPIOVAL	Read/write GPIO pin level (**applicable to VT-SOM-AH (6108) only**).
AT+ADC	Get the voltage value for an ADC channel.
AT+FWUPDATE	Set firmware update information (for firmware upgrade only).
AT+FWBINDL	Download binary firmware to FLASH (for firmware upgrade only).
+BEVENT	Unsolicited event notification.

2.3.1 AT

Query the status of the AT command interface.

AT Command	Expected Response
AT	OK

2.3.2 ATE

Control command echo.

AT Command	Expected Response
Disable echo: ATE0	OK
Enable echo: ATE1	OK

2.3.3 ATZ

Restore factory configuration parameters.

AT Command	Expected Response
ATZ	OK

2.3.4 AT+VER

Query device firmware and software package versions.

AT Command	Expected Response	Parameter Description
AT+VER?	+VER:"firmware version","morselib version","model" OK	<firmware version>: Firmware version <morselib version>: Morse Lib version <model>: Device model
Example	AT+VER? +VER:"V1.2.2","V1.17.6","MM8108" OK	

2.3.5 AT+RST

Reset the device.

AT Command	Expected Response
AT+RST	OK

2.3.6 AT+BOOT

Query the cause of the last system reset.

AT Command	Expected Response	Cause Codes
AT+BOOT?	+BOOT:<reason> OK	"POR": Power on reset "WDT": Watchdog reset "SW": Software reset
Example	AT+BOOT? +BOOT:"SW" OK	

2.3.7 AT+UART

Configure UART baud rate and related parameters.

AT Command	Expected Response	Parameter Values
Set the parameters: AT+UART=<baud_rate>,<data_bits>,<stop_bits>,<parity>,<HFC>	OK	<baud_rate>: 1200,2400,4800, 9600, 19200, 38400, 57600, 115200 <data_bits>: 7, 8, 9 <stop_bits>: 1, 2 <parity>: 0 = No parity; 1 = Even parity; 2 = Odd parity <HFC>: 0 = RTS/CTS disabled; 1 = RTS/CTS enabled
Query the configuration: AT+UART?	+UART:<baud_rate>,<data_bits>,<stop_bits>,<parity>,<HFC> OK	
Example	AT+UART=115200,8,1,0,0 OK AT+UART? +UART:115200,8,1,0,0 OK	

2.3.8 AT+GPIOCONF

Configure GPIO pin direction and pull-up/down selection. This command is applicable to VT-SOM-AH (6108) only.

AT Command	Expected Response	Parameter Values
Set the parameters: AT+GPIOCONF=<number>,<direction>,<pull-up>	OK	<number>: GPIO pin number: 0~9 <direction>: 0 = Input; 1 = Output <pull-up>: 0 = Pull-down; 1 = Pull-up; 2 = No pull
Query the configuration: AT+GPIOCONF?	+GPIOCONF:<number>,<direction>,<pull-up> OK	
Example	AT+GPIOCONF=0,1,2 OK AT+GPIOCONF? +GPIOCONF:0,1,2 OK	

2.3.9 AT+GPIOVAL

Configure GPIO pin level. This command is applicable to VT-SOM-AH (6108) only.

Note: If a GPIO pin is configured as input, its level cannot be set.

AT Command	Expected Response	Parameter Values
Set the parameters: AT+GPIOVAL=<number>,<level>	OK	<number>: GPIO pin number: 0~9 <level>: 0 = Low level; 1 = High level
Query the configuration: AT+GPIOVAL?	+GPIOVAL:<number>,<level> OK	
Example	AT+GPIOVAL=0,1 OK AT+GPIOVAL? +GPIOVAL:0,1 OK	

2.3.10 AT+ADC

Retrieve the ADC channel voltage reading.

AT Command	Expected Response	Parameter Values
Query all channel configuration: AT+ADC?	+ADC:<channel>,<value> … OK	<channel>: 0-1 <value>: 0-3300 (unit: mV)
Query configuration of a specified channel: AT+ADC?=<channel>	+ADC:<channel>,<value> OK	
Example	AT+ADC? +ADC:0,1024 +ADC:1,1024 OK	

2.3.11 AT+FWUPDATE

Set firmware update parameters.

AT Command	Expected Response	Parameter Description
AT+FWUPDATE=<length>,<crc>	OK	<length>: Firmware size, in bytes <crc>: 32-bit hexadecimal value
Example	AT+FWUPDATE=802416,0xEE7671B7 OK	

2.3.12 AT+FWBINDL

Download firmware image to FLASH.

AT Command	Expected Response	Parameter Description
AT+ FWBINDL=<offset>,<length>,<crc>	OK	<offset>: Data offset <length>: Data length, fixed at 4096 bytes <crc>: 32-bit hexadecimal value
Example	AT+FWBINDL=0,4096,0xEE7671B7 OK <data> +BEVENT:"FWBINDL_DONE",0,4096 **or** +BEVENT:"FWBINDL_FAIL",0,4096	

2.3.13 +BEVENT

Unsolicited event notification. It is reported automatically by the module without requiring an explicit command.

Example	**Firmware download indication:** +BEVENT:"FWBINDL_DONE",0,4096 +BEVENT:"FWBINDL_FAIL",0,4096 **Device mode indication:** +BEVENT:"AT_MODE" +BEVENT:"DTU_MODE"

2.4 Wi-Fi HaLow Commands

Command	Description
AT+WMACADDR	Configure the module's MAC address.
AT+WCOUNTRY	Configure the country code.
AT+WSCAN	Scan for available HaLow SSIDs.
AT+WSCANSSID	Scan for a specified HaLow SSID.
AT+WSSID	Connect to a preconfigured HaLow AP in DTU Transmission Mode.
AT+WCONN	Connect to a HaLow AP.
AT+WDISCONN	Disconnect from a HaLow AP.
AT+WIPADDR	Configure IPv4 address.
AT+WDHCP	Enable or disable DHCP.
AT+WDNS	Configure DNS server.
AT+WPING	Execute 'ping' test.
AT+WIPERF	Execute 'iPerf' test.
AT+WTIMEOUT	Configure response timeout for specific commands.
+WEVENT	HaLow-related unsolicited event notification.

2.4.1 AT+WMACADDR

Query the module's MAC address.

AT Command	Expected Response	Parameter Description
AT+WMACADDR?	+WCOUNTRY:"mac" OK	<mac>: MAC address, Hexadecimal string (e.g., 01:02:03:04:05:06)
Example	AT+WMACADDR? +WMACADDR:"01:02:03:04:05:06" OK	

The MAC address in the example is for illustrative only.

2.4.2 AT+WCOUNTRY

Configure the module's country code.

AT Command	Expected Response	Parameter Description
Set the country code: AT+WCOUNTRY=<country>	OK	<country>: Country code for Wi-Fi HaLow operation
Query the country code: AT+WCOUNTRY?	+WCOUNTRY:"country " OK	
Example	AT+WCOUNTRY=US OK AT+WCOUNTRY? +WCOUNTRY:"US" OK	

Supported country codes are listed below.

Code	Country	Code	Country	Code	Country
AU	Australia	**CA**	Canada	**EU**	Europe
GB	United Kingdom	**IN**	India	**JP**	Japan
KR	South Korea	**NZ**	New Zealand	**US**	USA

2.4.3 AT+WSCAN

Scan for available Wi-Fi HaLow SSIDs.

AT Command	Expected Response
AT+WSCAN	+WSCAN:"bssid",frequency@bandwidth,"rssi","security","ssid" OK
Example	AT+WSCAN +WSCAN:"18:9b:a5:18:10:11",907.5@8,-31,"OWE","DGL-AH-101-1011" +WSCAN:"18:9b:a5:b6:a6:04",907.5@8,-73,"SAE","DGL-AH-101-A604" +WSCAN:"fc:70:2e:0c:64:27",916.5@2,-59,"SAE","ssid-sae-4" +WSCAN:"18:9b:a5:15:09:12",907.0@8,-34,"SAE","DGL-AH-101-0912" OK

2.4.4 AT+WSCANSSID

Scan for a specified HaLow AP SSID.

AT Command	Expected Response
AT+WSCANSSID=<ssid>	+WSCANSSID: "bssid",frequency@bandwidth,"rssi","security","ssid" OK
Example	AT+WSCANSSID=”ekh01-9762” +WSCANSSID:"0c:bf:74:06:97:62",915.0@8,-58,"SAE","ekh01-9762" OK

In this command, <ssid> specifies the target HaLow AP SSID. It is a string with a maximum of 32 characters.

2.4.5 AT+WSSID

Configure the HaLow AP parameters for the module to establish a HaLow connection in **DTU Transmission Mode**.

AT Command	Expected Response	Parameter Description
Configure the HaLow AP parameters: AT+WSSID=<ssid>,<security>[,<password>]	OK	<ssid>: Target HaLow AP SSID <security>: Encryption mode: open, owe, sae (default) [<password>]: Not applicable for “open” and “owe” security modes
Query the configuration: AT+WSSID?	+WSSID:"ssid","security","password" OK	
Example	AT+WSSID="ekh01-9762","sae","12345678" OK AT+WSSID? +WSSID:"ekh01-9762","sae","12345678" OK	

2.4.6 AT+WCONN

Connect to a HaLow AP in AT Command Mode.

AT Command	Expected Response	Parameter Description
Connect to target HaLow AP: AT+WCONN=<ssid>,<security>[,<password>]	OK	<ssid>: Target HaLow AP SSID <security>: Encryption mode: open, owe, sae (default) [<password>]: Not applicable for “open” and “owe” security modes <state>: Connection state: disconnected, connecting, connected
Query the connection status: AT+WCONN?	+WCONN:"ssid","security","password","state" OK	
Example	AT+WCONN="ekh01-9762","sae","12345678" OK AT+WCONN? +WCONN:"ekh01-9762","sae","12345678","connected" OK	

2.4.7 AT+WDISCONN

Disconnect the module from the current HaLow AP.

AT Command	Expected Response
AT+WDISCONN	OK
Example	AT+WDISCONN OK +WEVENT:"DISCONNECT","ekh01-9762","sae"

2.4.8 AT+WIPADDR

Assign a **static** IP address to the module (applicable only when DHCP is disabled).

AT Command	Expected Response	Parameter Description
Set a static IP address: AT+WIPADDR=<ip>,<netmask>,<gateway>	OK	<ip>: Module's IP address (e.g., 192.168.4.197) <netmask>: Subnet mask (e.g., 255.255.255.0) <gateway>: Gateway IP (e.g., 192.168.4.1)
Query the module's IP address: AT+WIPADDR?	+WIPADDR:"ip","netmask","gateway" OK	
Example	AT+WIPADDR="172.18.1.197","255.255.255.0","172.18.4.1" OK AT+WIPADDR? +WIPADDR:"172.18.1.197","255.255.255.0","172.18.4.1" OK	

1. When **DHCP** is enabled, the IP address obtained from the HaLow AP will be released upon disconnection. When **DHCP** is disabled and a static IP is configured, the IP address remains unchanged even if the AP connection is lost.
2. After modifying the static IP configuration, a device reboot is required for the new settings to take effect.

2.4.9 AT+WDHCP

Enable or disable DHCP service.

AT Command	Expected Response	Parameter Description
Enable/disable DHCP: AT+WDHCP=<mode>	OK	<mode>: DHCP mode 0 = DHCP disabled; 1 = DHCP enabled (default)
Query the DHCP status: AT+WDHCP?	+WDHCP:mode OK	
Query the module's IP address: AT+WIPADDR?	+WIPADDR:"ip","netmask","gateway" OK	<ip>: Module's IP address (e.g., 192.168.4.197) <netmask>: Subnet mask (e.g., 255.255.255.0) <gateway>: Gateway IP (e.g., 192.168.4.1)
Example	AT+WDHCP? +WDHCP:1 OK	

Changing the DHCP mode requires a device reboot to take effect.

2.4.10 AT+WDNS

Configure DNS server addresses.

AT Command	Expected Response	Parameter Description
Set the DNS server address: AT+WDNS=<dns1>[,<dns2>]	OK	<dns1>: Primary DNS server IP [<dns2>]: Secondary DNS server IP (optional) Example IP: 192.168.1.1
Query the DNS server address: AT+WDNS?	+WDNS:"dns1","dns2" OK	
Example	AT+WDNS="192.168.1.1" OK AT+WDNS? +WDNS:"192.168.1.1" OK	

2.4.11 AT+WPING

Execute a ping test.

AT Command	Expected Response	Parameter Description
AT+WPING=<ip>,<count>	+WPING:64,"ip",count,round_trip OK	<ip>: Target IP address for the test (e.g., 192.168.1.1) <count>: Number of ping requests
Example	AT+WPING="192.168.1.1",10 +WPING:64,"192.168.1.1",1,182 +WPING:64,"192.168.1.1",2,44 +WPING:64,"192.168.1.1",3,18 +WPING:64,"192.168.1.1",4,13 +WPING:64,"192.168.1.1",5,13 +WPING:64,"192.168.1.1",6,13 +WPING:64,"192.168.1.1",7,13 +WPING:64,"192.168.1.1",8,13 +WPING:64,"192.168.1.1",9,13 +WPING:64,"192.168.1.1",10,13 OK +WEVENT:"PING_STATS","192.168.1.1",10 packets transmitted,10 packets received,0% packet loss,round-trip min/avg/max = 13/40/182 ms	

1. The module must connect to a HaLow AP before running a ping test.
2. Sending 'AT\r' during a ping test will abort the operation.

2.4.12 AT+WIPERF

Run an iPerf performance test.

AT Command	Expected Response	Parameter Description
AT+ WIPERF=<mode>,<port> [,<ip> [,<count>]]	OK	<mode>: iPerf operating mode (TCP_SERVER, UDP_SERVER, TCP_CLIENT, UDP_CLIENT) <port>: Port number (1-65535, typically 5001) [<ip>]: Remote server IP (required for client modes; e.g., 192.168.1.1) [<count>]: Number of test packet (client mode only)
Example	• TCP Server: AT+WIPERF="TCP_SERVER",5001 • UDP Server: AT+WIPERF="UDP_SERVER",5001 • TCP Client: AT+WIPERF="TCP_CLIENT",5001, "172.18.4.1",10 • UDP Client: AT+WIPERF="UDP_CLIENT",5001, "172.18.4.1",10	

The module must connect to a HaLow AP before running an iPerf test.

2.4.13 AT+WTIMEOUT

Configure timeout for specific commands.

AT Command	Expected Response	Parameter Description
Set the timeout: AT+ WTIMEOUT=<command>, <timeout>	OK	<command>: Target command. Currently supported: WCONN <timeout>: Timeout duration. Value: 1-60 (seconds); default: 60
Query the timeout settings: AT+ WTIMEOUT?	+WTIMEOUT:"command",timeout OK	
Example	AT+WTIMEOUT="WCONN",60 OK AT+WTIMEOUT? +WTIMEOUT:"WCONN",60 OK	

2.4.14 +WEVENT

HaLow-related unsolicited event notification.

Example	**Connectivity related**: +WEVENT:"CONNECTING"," ekh01-9762","sae" +WEVENT:"CONNECT_SUCCESS"," ekh01-9762","sae" +WEVENT:"DISCONNECT"," ekh01-9762","sae" +WEVENT:"CONNECT_TIMEOUT"," ekh01-9762","sae" **Ping test related**: +WEVENT:"PING_STATS","192.168.1.1",10 packets transmitted,10 packets received,0% packet loss,round-trip min/avg/max = 13/40/182 ms **iPerf test related**: +WEVENT:"IPERF_TCP_SERVER",… +WEVENT:"IPERF_UDP_SERVER",… +WEVENT:"IPERF_TCP_CLIENT",… +WEVENT:"IPERF_UDP_CLIENT",…

2.5 Socket Commands

Command	Description
AT+SOCKET	Configure remote socket parameters of the target device for communication in DTU Transmission Mode.
AT+SOPEN	Create a UDP/TCP socket.
AT+SCLOSE	Close an existing socket.
AT+SLIST	List all open sockets.
AT+SSEND	Send data through a specified socket.
AT+SRECV	Read received data.
AT+SRECVMODE	Configure data receive mode.
AT+SRECVINFO	Configure received data display mode.
AT+STCPKEEPALIVE	Configure TCP Client keep-alive parameters.
AT+STCPNODELAY	Configure TCP Nagle algorithm state.
+RXD	Data reception indication.
+SEVENT	Socket-related unsolicited event notification.

2.5.1 AT+SOCKET

Configure remote socket parameters of the target device for communication in **DTU Transmission Mode**.

AT Command	Expected Response	Parameter Description
Configure the target device parameters: AT+SOCKET=<type>,<ip>,<port>	OK	<type>: Communication type: UDP or TCP. In TCP mode, the module acts only as a TCP Client <ip>: IP address of the target device (e.g., 192.168.1.1) <port>: Listening port of the target device
Query the configuration: AT+SOCKET?	+SOCKET:"type","ip",port OK	
Example	**Configure for UDP communication:** AT+SOCKET="UDP","192.168.1.1",10000 OK **Configure for TCP communication:** AT+SOCKET="TCP","192.168.1.1",10000 OK AT+SOCKET? +SOCKET:"UDP","192.168.1.1",10000 OK	

2.5.2 AT+SOPEN

Create a socket.

AT Command	Expected Response	Parameter Description
Create a UDP socket: AT+SOPEN=<UDP>,<local_port>	+OPEN:<sock> OK	<sock>: Socket ID (0–3) upon successful creation <local_port>: Local port number of local device <remote_ip>: IP address of the target device (e.g., 192.168.1.1) <remote_port>: Port number of the target device
Create a TCP Server socket: AT+SOPEN=<TCP>,<local_port>		
Create a TCP Client socket: AT+SOPEN=<TCP>,<remote_ip>,<remote_port>		
Example	**Create a UDP socket:** AT+SOPEN="UDP",10000 +SOPEN:0 OK **Create a TCP client socket:** AT+SOPEN="TCP","172.18.1.221",10000 +SOPEN:1 OK **Create a TCP server socket:** AT+SOPEN="TCP",10000 +SOPEN:2 OK	

1. The device supports up to 4 sockets.
2. Creating a TCP Server consumes 2 sockets, because when a TCP Client connects to the Server, the Server allocates a new socket for that Client.

2.5.3 AT+SCLOSE

Close sockets.

AT Command	Expected Response	Parameter Values
Close all sockets: AT+SCLOSE	OK	/
Close a specified socket: AT+SCLOSE=<sock>	OK	<socket>: Socket ID
Example	**Close all sockets:** AT+SCLOSE OK **Close socket 0:** AT+SCLOSE=0 OK	

2.5.4 AT+SLIST

List all open sockets.

AT Command	Expected Response	Parameter Values
Query all open sockets: AT+SLIST?	+SLIST:<sock>,<port>,<state> … OK	<sock>: Socket ID <port>: Port number <state>: Socket creation status. 0 = Actively created 1 = Passively created
Example	AT+SLIST? +SLIST:0,"UDP",1000,0 +SLIST:1,"TCP",10000,1 +SLIST:2,"TCP",50,0 OK	

2.5.5 AT+SSEND

Send data to the target device over a socket.

<table>
<tr><th>AT Command</th><th>Expected Response</th><th>Parameter Description</th></tr>
<tr><td>Send data to a UDP device:
AT+SSEND=<sock>,<remote_ip>,<port>,<data_size>,<done_event></td><td rowspan="2">OK
<data>
+SEVENT:"SEND_DONE",<data_size></td><td rowspan="2"><sock>: Socket ID
<remote_ip>: Destination IP (e.g., 192.168.1.1)
<port>: Port of the target device
<data_size>: Data bytes
<done_event>: Event notification for data transmission result (0-1)
0 = Notification disabled
1 = Notification enabled
<data>: Data transmitted</td></tr>
<tr><td>Send data to a TCP Server/Client:
AT+SSEND=<sock>,<data_size>,<done_event></td></tr>
<tr><td>Example</td><td colspan="2">Send data to a UDP device:
AT+SSEND=0,"172.18.1.221",10000,11,1
OK
Hello World
+SEVENT:"SEND_DONE",11

Send data to a TCP Server/Client:
AT+SSEND=0,11,1
OK
Hello World
+SEVENT:"SEND_DONE",11</td></tr>
</table>

2.5.6 AT+SRECV

Read data received from a specified socket.

AT Command	Expected Response	Parameter Description
AT+SRECV=<sock>,<size>	<msg>	<sock>: Socket ID <size>: Received data bytes <msg>: Received data

1. When using this command to read the received data, first set AT+SRECVMODE=1 to enable passive receive mode.
2. Data can only be read once. The buffer is cleared after each read and waits for new data.
3. The module issues a +RXD notification upon receiving data. If not read promptly, the data will be overwritten by subsequent incoming data.

2.5.7 AT+SRECVMODE

Configure data receiving mode.

AT Command	Expected Response	Parameter Description
Set the mode: AT+SRECVMODE=<mode>	OK	<mode>: Data receiving mode. 0 = Active mode (default) 1 = Passive mode
Query current mode: AT+SRECVMODE?	+SRECVINFO:<mode> OK	
Example	AT+SRECVMODE=1 OK AT+SRECVNODE? +SRECVMODE:1 OK	

1. Active Mode: When the device receives data, it will notify via +RXD and simultaneously output the received data.
2. Passive Mode: When the device receives data, it will notify via +RXD. The received data must be read using AT+SRECV.

2.5.8 AT+SRECVINFO

Configure received data display mode.

AT Command	Expected Response	Parameter Description
Configure display mode for a specified socket: AT+SRECVINFO=<sock>,<mode>	OK	<sock>: Socket ID <mode>: Display mode for received data (0-1). 0 = Concise mode 1 = Verbose mode (default)
Query display mode of all sockets: AT+SRECVINFO?	+SRECVINFO:<sock>,<mode> … OK	
Query display mode of a specified socket: AT+SRECVINFO?=<sock>	+SRECVINFO:<sock>,<mode> OK	

2.5.9 AT+STCPKEEPALIVE

Configure TCP Client keep-alive parameters.

AT Command	Expected Response	Parameter Value
Set the parameters: AT+STCPKEEPALIVE=<s>,<keepalive>[,<keepidle>[,<keepcnt>[,<keepintvl>]]]	OK	<s>: TCP Client socket ID <keepalive>: Keep-alive status (0-1). 0 = Disable, 1 = Enable [<keepidle>]: Idle detection interval (unit: seconds) [<keepcnt>]: Probe transmission count [<keepintvl>]: Probe transmission interval (unit: seconds)
Query the parameters: AT+STCPKEEPALIVE?	AT+STCPKEEPALIVE=<s>,<keepalive>[,<keepidle>[,<keepcnt>[,<keepintvl>]]] OK	
Example	AT+STCPKEEPALIVE=1,1,10,5,3 OK AT+STCPKEEPALIVE? +STCPKEEPALIVE:0,0 +STCPKEEPALIVE:1,1,10,5,3 … OK	

2.5.10 AT+STCPNODELAY

Configure **TCP** Nagle algorithm state.

AT Command	Expected Response	Parameter Value
Set the state: AT+STCPNODELAY=<sock>,<state>	OK	<sock>: TCP socket ID <state>: Nagle algorithm state. 0 = Disable (default) 1 = Enable
Query the state: AT+STCPKEEPALIVE?	+STCPNODELAY:<sock>,<state> … OK	
Example	AT+STCPNODELAY =0,0 OK AT+STCPNODELAY? +STCPNODELAY:0,0 … OK	

Nagle algorithm is only applicable to TCP connections.

2.5.11 +RXD

Data reception indication.

Mode	Description
1. Active Verbose Mode: +RXD:<sock>,<data_size>,<ip>,<port> <data> 2. Active Concise Mode: +RXD:<sock>,<data_size> <data> 3. Passive Verbose Mode: +RXD:<sock>,<data_size>,<ip>,<port> 4. Passive Concise Mode: +RXD:<sock>,<data_size>	<sock>: Socket ID <data_size>: Received data bytes <ip>: Source IP of the data <port>: Port number <data>: Received data
Example	**Active Verbose Mode:** +RXD:0,21,"172.18.1.221",10000 Hello,my name is STA2 **Passive Verbose Mode:** +RXD:0,21,"172.18.1.221",10000 AT+SRECV=0,21 Hello,my name is STA2

2.5.12 +SEVENT

Socket-related unsolicited event notification.

AT Command	Parameter Description
+SEVENT:<code>,<sock>[,<data_size>]	<code>: Information prompt string <sock>: Socket ID <data_size>: Data bytes
Example	**For TCP client connection events, TCP Client ID is the socket ID:** +SEVENT:"CONNECT_OK",1 +SEVENT:"CONNECT_FAIL",1 +SEVENT:"DISCONNECT",1 **For TCP server connection events, TCP Client ID is the socket ID:** +SEVENT:"CONNECTED",1 **For TCP and UDP data sending events:** +SEVENT:"SEND_DONE",1 +SEVENT:"SEND_FAIL",1

CHAPTER 3 CONNECT TO A HALOW AP

3.1 Connection in AT Command Mode

By default, the modules do not automatically connect a HaLow access point (AP). Manual connection via the AT+WCONN command is required.

Vantron offers a range of compatible HaLow units that can function as the HaLow AP, including the HAP101, HAP103, HAP202, and BR101.

The parameters required for the modules to establish HaLow connectivity include the AP's SSID, encryption method (security), password, and country code.

In this document, the HaLow AP used has the following parameters:

Country: US

SSID: DGL-AH-101-A513

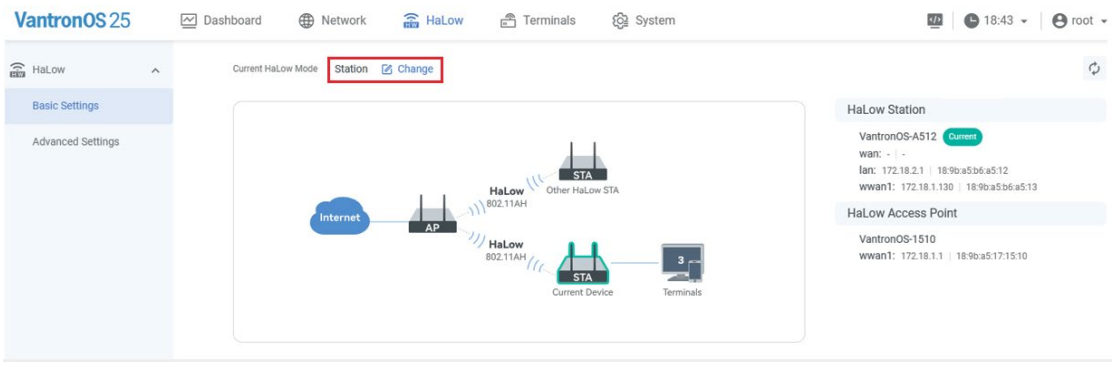
Encryption Mode (Security): WPA3-SAE

Password: 12345678

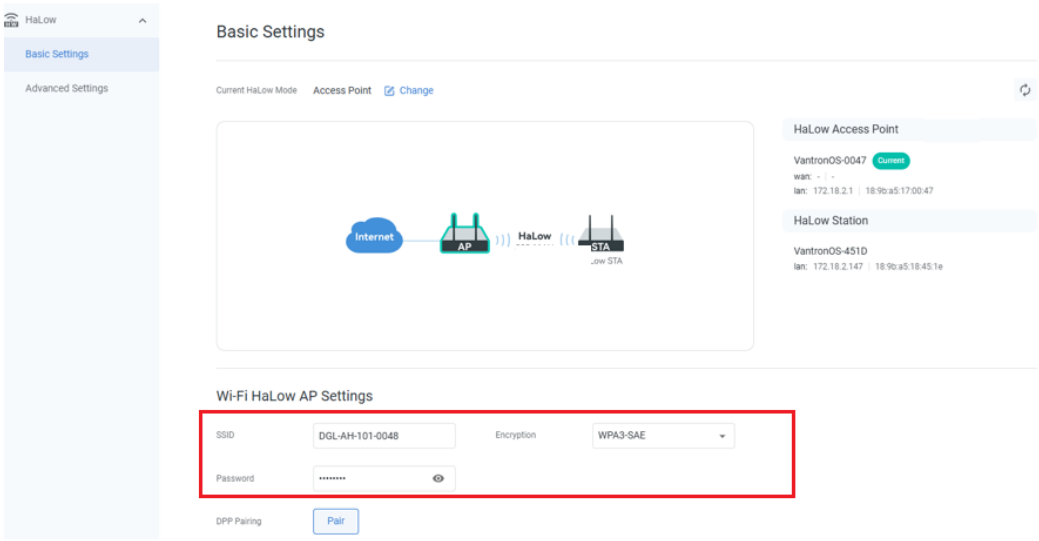
3.1.1 AP Setup

For Vantron HaLow units that default to HaLow **Station** mode, you may need to refer to the corresponding user manual to log in to the device for corresponding AP information.

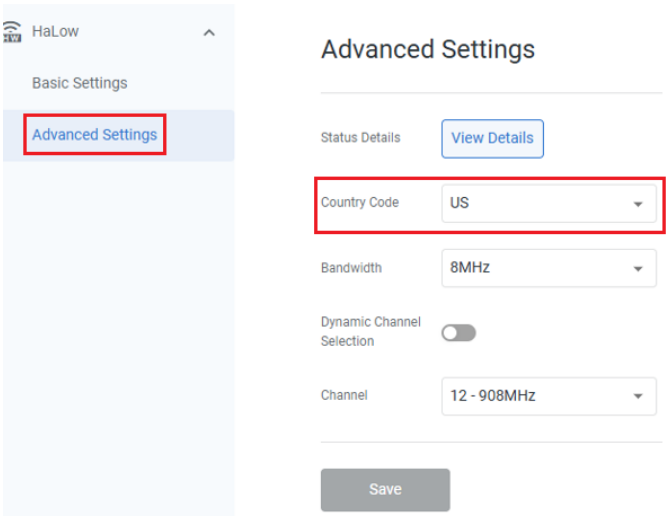
- The screenshot below illustrates how to switch the HaLow operation mode after logging in to the device.



- The screenshot below highlights the fields for the HaLow AP's SSID, encryption method, and password.



- Country code is available in **Advanced Settings**.



The above screenshots are for illustration purposes only. Refer to your AP's user manual for device login and specific configuration details.

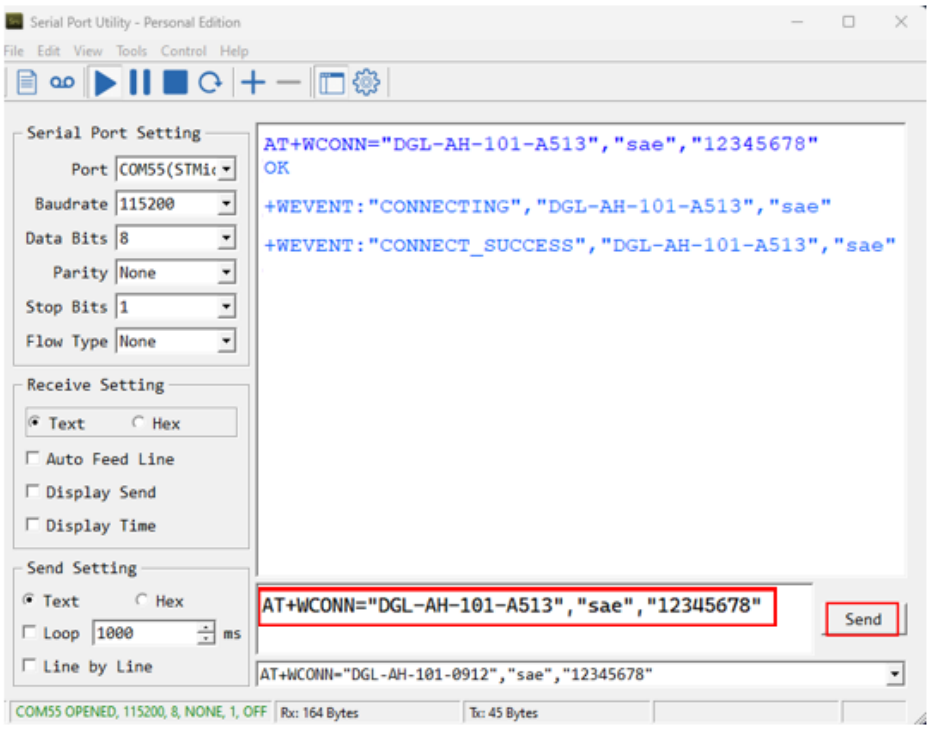
3.1.2 STA Setup

The modules support operation **only as a HaLow Station (STA)**.

The default country code for the modules is set to **US**. If it does not match the country code configured on the HaLow AP, use the AT+COUNTRY command to modify it. After each configuration change, execute the ATZ command or power-cycle the module to apply the changes.

After the HaLow AP is ready, send the following command to the module via UART/USB to establish a connection: AT+WCONN=<ssid>,<security>[,<password>].

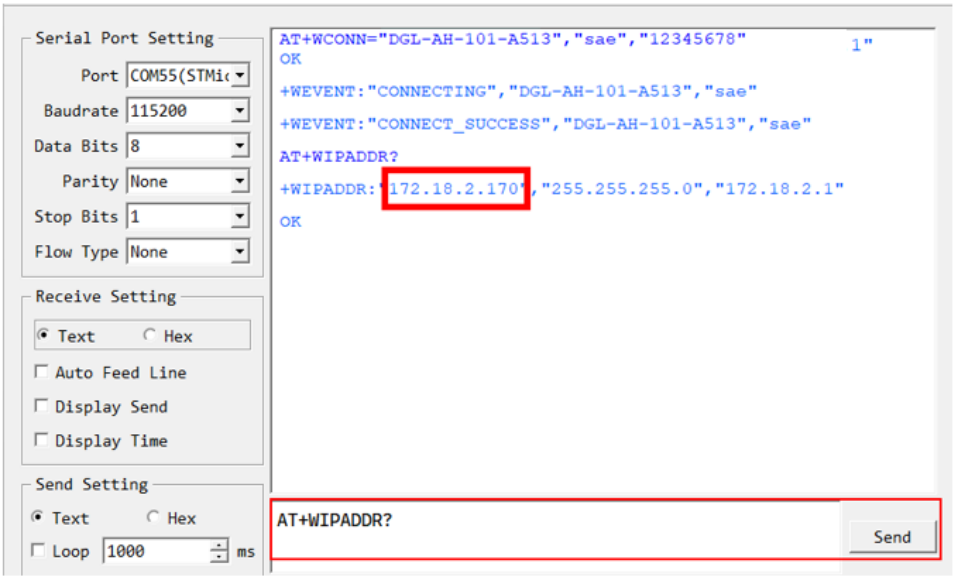
In this document: AT+WCONN="DGL-AH-101-A513","sae","12345678"



Upon successful HaLow connection, the following unsolicited notification will appear.

```
AT+WCONN="DGL-AH-101-A513","sae","12345678"
OK

+WEVENT:"CONNECTING","DGL-AH-101-A513","sae"
+WEVENT:"CONNECT_SUCCESS","DGL-AH-101-A513","sae"
```

The modules have DHCP enabled by default; an IP address will be automatically assigned to the module upon connection. Use the AT+WIPADDR? command to query the information.



3.2 Connection in DTU Transmission Mode

DTU Transmission Mode operates automatically once module parameters are configured in AT Command Mode. After completing configuration, switch to DTU Transmission Mode and power-cycle the module to apply all settings.

The procedure for configuring DTU mode parameters is as follows:

1. Enter AT Command Mode (skip this step if the module is already in this mode).

```
+++++
```

2. Configure the country code.

```
AT+WCOUNTRY=<country>
```

3. Configure DHCP mode.

```
AT+WDHCP=<mode>                    // 1 = DHCP enabled, 0 = Static IP mode
```

4. Configure parameters of the target HaLow AP for a HaLow connection.

```
AT+WSSID=<ssid>,<security>[,<password>]
```

5. If DHCP in step 3 is set to 0, configure the static IP address.

```
AT+WIPADDR=<ip>,<netmask>,<gateway>
```

6. Query the module's IP address.

```
AT+WIPADDR?
```

7. Configure remote Socket parameters of the target device for module communication.

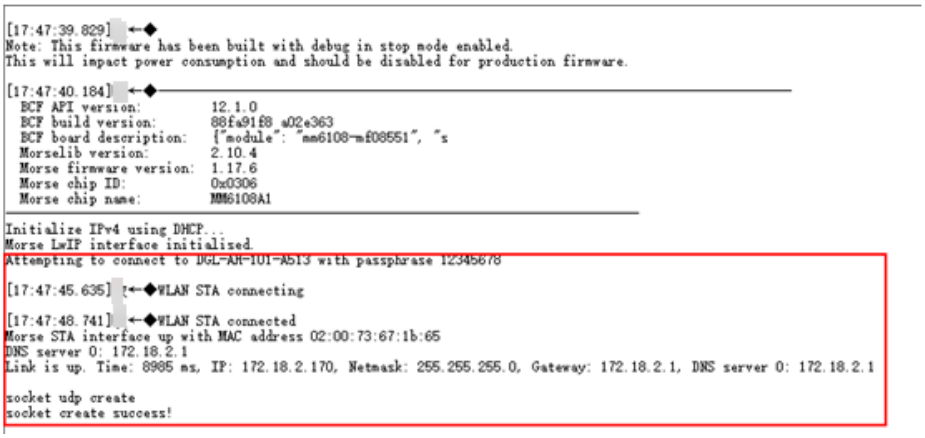
```
AT+SOCKET=<type>,<ip>,<port>
```

8. Exit AT command mode and switch to DTU Transmission Mode.

```
+++++
```

9. Power-cycle the module to apply the settings.

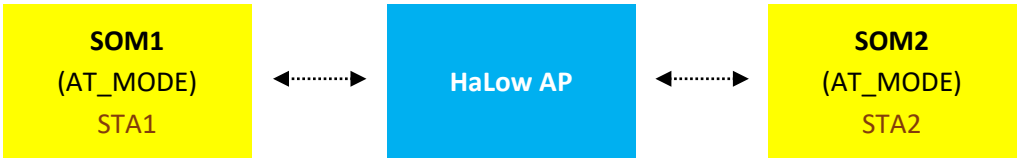
After completing above steps, log messages will be printed from the module's Console port.



CHAPTER 4 DATA COMMUNICATION

4.1 Communication in AT Command Mode

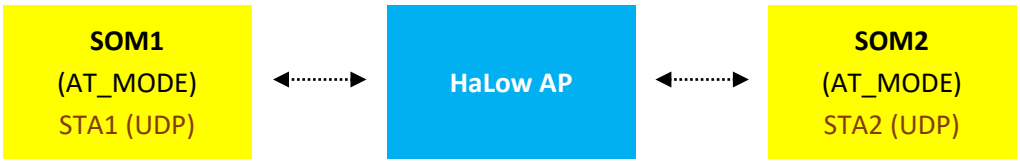
In AT Command Mode, one module can communicate with another via a HaLow AP. The corresponding topology is as follows:



Ensure both STA1 and STA2 are connected to the same HaLow AP before initiating data communication.

4.1.1 UDP Data Communication

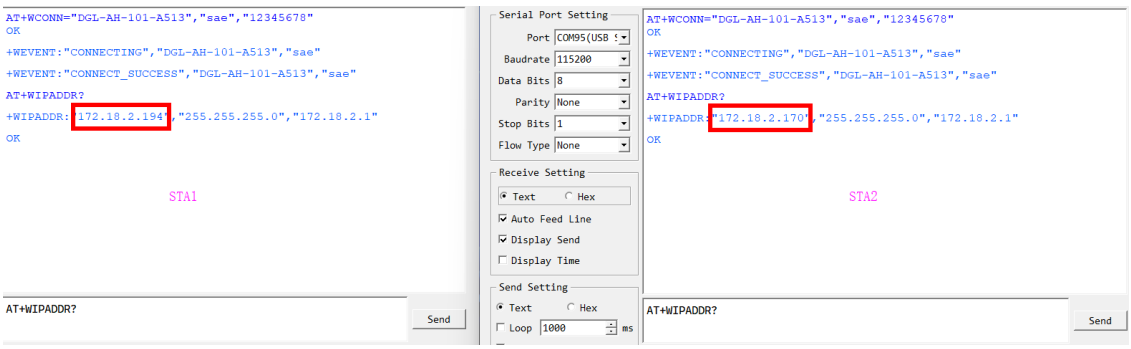
Example: AT-Mode UDP Data Communication Between Two SOMs.



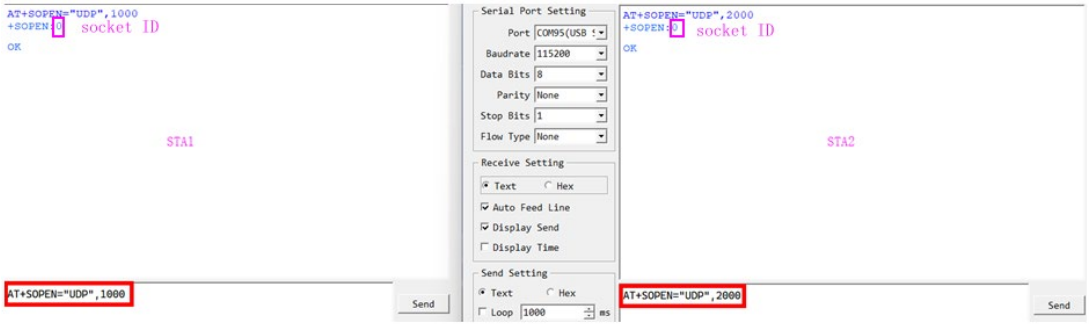
1. Connect STA1 and STA2 to the same AP and obtain their corresponding IP addresses using the AT+WIPADDR? command after successful connection.

   STA1: 172.18.2.194

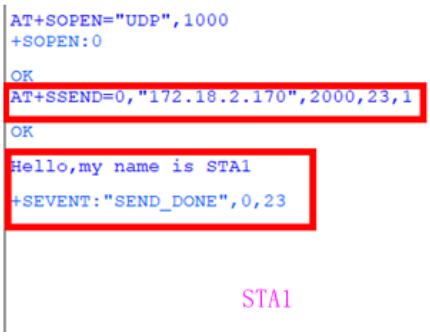
   STA2: 172.18.2.170



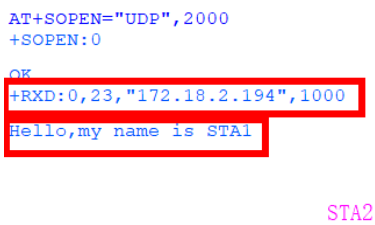
2. Create a UDP socket on STA1 and STA2 respectively using the AT+SOPEN command.
   - For instance, set the local port of STA1 to 1000 and that of STA2 to 2000
   - Valid port range: 1-65535



3. Suppose **STA1** sends the string “Hello,my name is STA1” to STA2, which is 21 bytes in length. When sending the data, carriage return and line feed (\r\n) are automatically appended. Therefore, the actual transmitted data length becomes 23 bytes.
   - Send: AT+SSEND= <sock>,<remote_ip>,<port>,<data_size>,<done_event>
   - Expected response: OK
   - Send: Hello,my name is STA1
   - Expected response: +SEVENT:"SEND_DONE",0,23



   - On STA2:
     1) If the data receiving mode and received data display mode are **not configured** via AT+RECVMODE and AT+SRECVINFO, the module defaults to **Active Verbose Mode** for data reception and display.



2) If STA2 is set to **Active Concise Mode**, the received data is displayed as follows:

```
AT+SOPEN="UDP",2000
+SOPEN:0

OK
+RXD:0,23,"172.18.2.194",1000

Hello,my name is STA1
AT+SRECVINFO=0,0

OK

+RXD:0,23

Hello,my name is STA1
```

3) If STA2 is set to **Passive Verbose Mode**, the received data is displayed as follows:

```
+RXD:0,23,"172.18.2.194",1000

Hello,my name is STA1
AT+SRECVINFO=0,0

OK

+RXD:0,23

Hello,my name is STA1
AT+SRECVINFO=0,1

OK

AT+SRECVMODE=1

OK

+RXD:0,23,"172.18.2.194",1000

AT+SRECV=0,23

Hello,my name is STA1
```

4) If STA2 is set to **Passive Concise Mode**, the received data is displayed as follows:

```
+RXD:0,23,"172.18.2.194",1000

Hello,my name is STA1
AT+SRECVINFO=0,0

OK

+RXD:0,23

Hello,my name is STA1
AT+SRECVINFO=0,0

OK

AT+SRECVMODE=1

OK

+RXD:0,23

Hello,my name is STA1
```

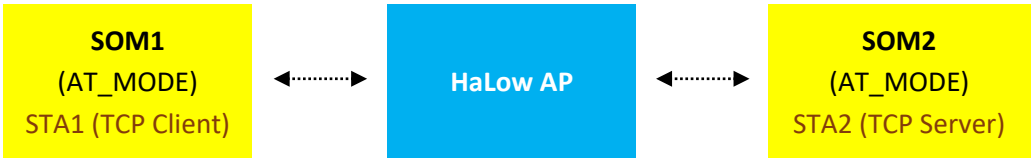
4. STA2 can also send data to STA1 in the same way.

   Suppose **STA2** sends the string “Hello,my name is STA2” to STA1, the execution steps are as follows:

   - Send: AT+SSEND=<sock>,<remote_ip>,<port>,<data_size>,<done_event>
   - Expected response: OK
   - Send: Hello,my name is STA2
   - Expected response: +SEVENT:"SEND_DONE",0,23
   - Refer to the prior step for the commands to display the receive data on STA1.

4.1.2 TCP Data Communication

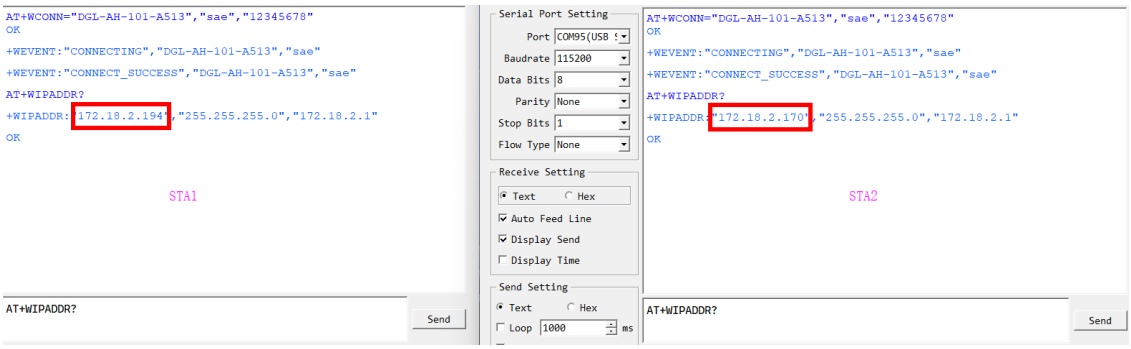
TCP communication requires designating one device as the TCP Client and another as the TCP Server. In this document, STA1 is defined as the TCP Client, and STA2 as the TCP Server.



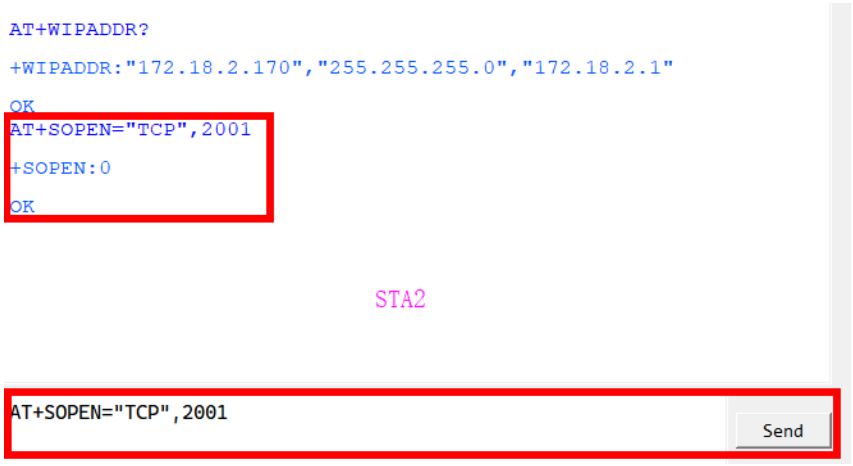
1. Connect STA1 and STA2 to the same AP and obtain their corresponding IP addresses using the AT+WIPADDR? command after successful connection.

   STA1: 172.18.2.194

   STA2: 172.18.2.170

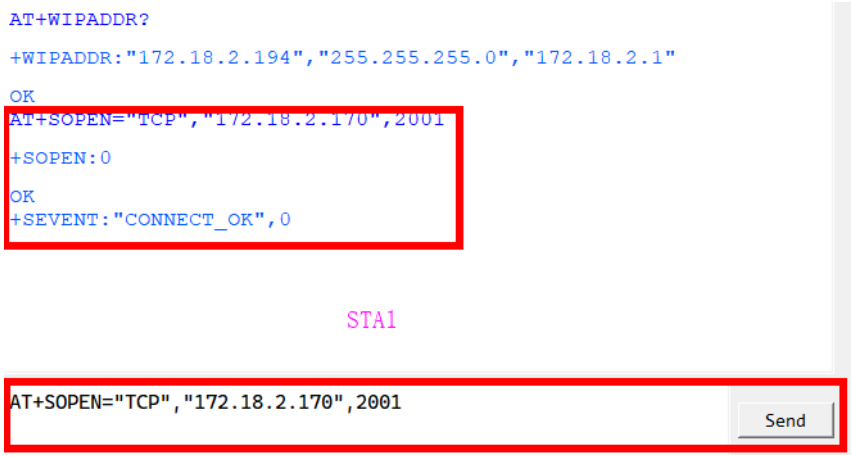


2. **On STA2**: Create a TCP Server socket using AT+SOPEN on port 2001. Once successfully created, the module waits for a TCP Client connection. When the module acts as a TCP Server, it supports only **one** TCP Client connection.



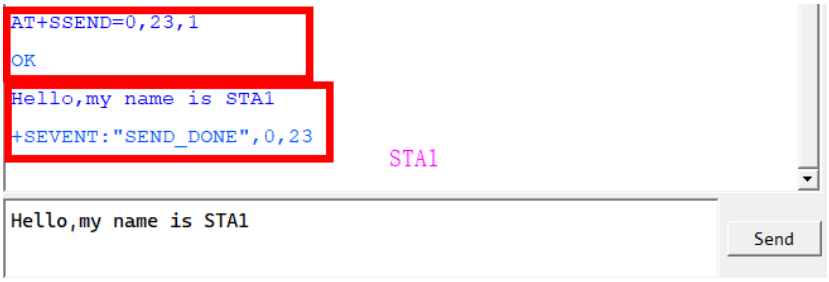
3. **On STA1**: Create a TCP Client socket using AT+SOPEN, and fill in the IP address and port number of the TCP Server during configuration.

   - If the TCP Server is already listening for connections upon successful socket creation, the TCP Client will connect to the TCP Server immediately, with the prompt: +SEVENT:"CONNECT_OK", <sock>, where <sock> indicates the socket ID of the connected TCP Server.

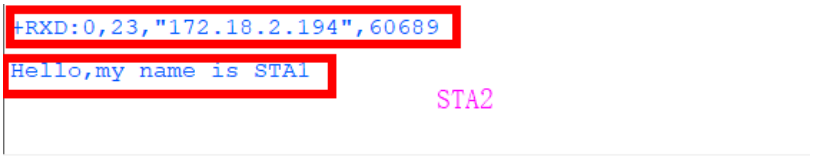


   - Meanwhile, STA2 will also output a connection event prompt: +SEVENT:"CONNECT_OK", <sock>, where <sock> indicates the socket ID of the connected TCP client.

4. During TCP communication, it is no longer necessary to fill in the peer IP address and port number when sending data via AT+SSEND. Because:
   - The TCP Server automatically records the IP address and port number of the TCP Client upon connection.
   - The TCP Client already records the IP address and port number of the TCP Server when the socket is created using AT+SOPEN.
5. Suppose **STA1** sends the string "Hello,my name is STA1" to STA2, which is 21 bytes long. When sending the data, carriage return and line feed (\r\n) are appended automatically. Hence the actual transmitted data length is 23 bytes.
   - Send: AT+SSEND=<sock>,<data_size>,<done_event>
   - Expected response: OK
   - Send: Hello,my name is STA1
   - Expected response: +SEVENT:"SEND_DONE",0,23



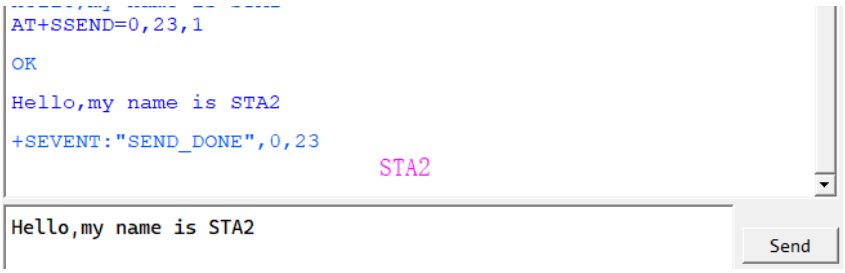
6. STA2 displays the data received.



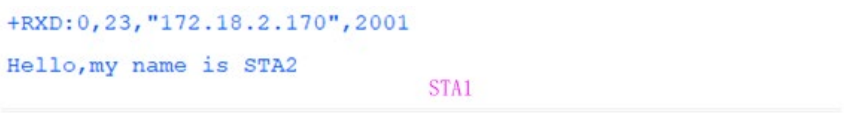
7. STA2 can also send data to STA1 in the same way.

   Suppose **STA2** sends the string **Hello,my name is STA2** to STA1, the execution steps are as follows:

   - Send: AT+SSEND=<sock>,<data_size>,<done_event>
   - Expected response: OK
   - Send: Hello,my name is STA2
   - Expected response: +SEVENT:"SEND_DONE",21



8. STA1 displays the data received.



Note:

If the data receiving mode and received data display mode are **not configured** via AT+RECVMODE and AT+SRECVINFO, the module defaults to **Active Verbose Mode** for data receiving and displaying.

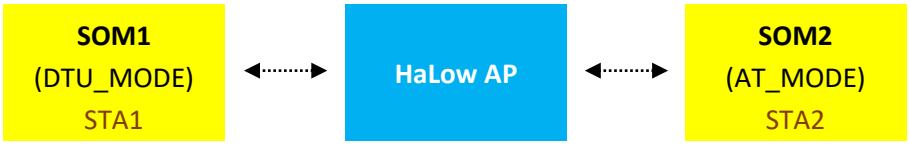
For related AT commands used to configure the module's reception and display modes after peer data transmission, refer to the UDP communication description in **Section 4.1.1**.

4.2 Communication in DTU Transmission Mode

The DTU Transmission Mode supports both UDP and TCP communication. For TCP communication, the module can act only as a **TCP Client**, not as a TCP Server. The communication mode is configured via the AT+SOCKET command in AT Command Mode. Refer to Section 3.2 Connection in DTU Transmission Mode for the DTU-mode parameter configuration instructions.

In DTU Transmission Mode, the module automatically connects to the HaLow AP after power-on without external intervention.

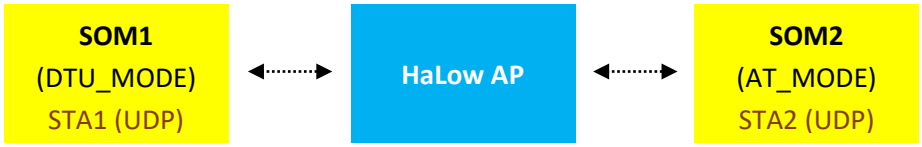
Typically, one module in DTU Transmission Mode communicates with another module in AT Command Mode through a HaLow AP. In a typical DTU-mode UDP communication scenario as follows, SOM1 (DTU mode) requires no further operation after parameter configuration in AT mode.



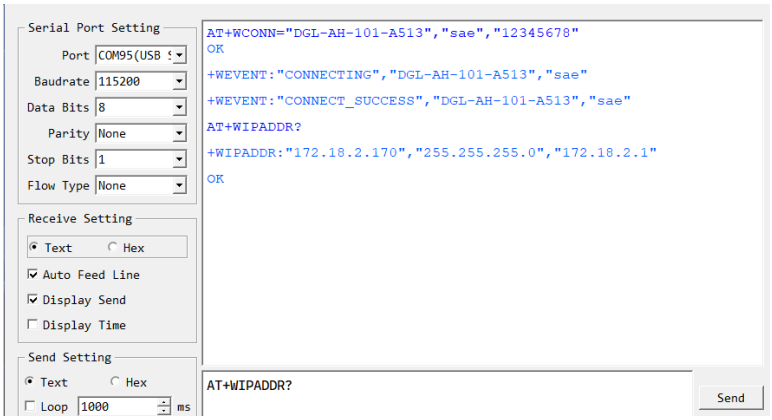
4.2.1 UDP Data Communication

Example: UDP Data Communication Between DTU-mode STA1 and AT-mode STA2.

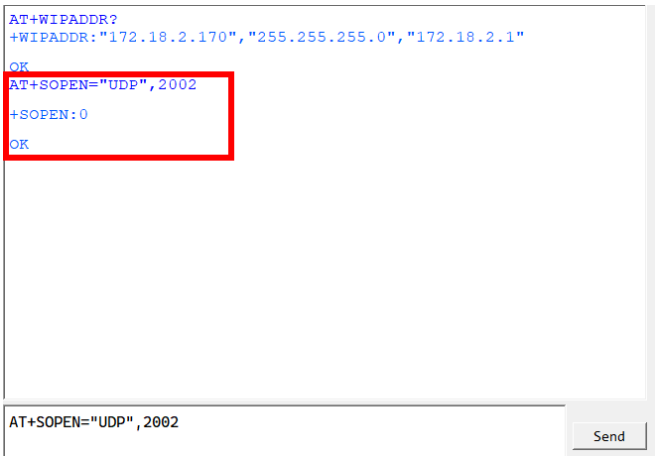
Configuration: STA1: Set to UDP mode; STA2: Create a UDP socket



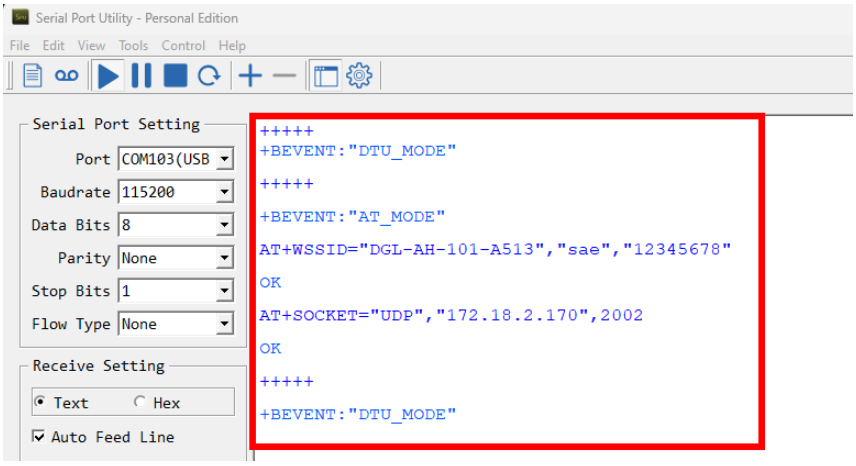
1. Connect STA2 to a HaLow AP, and query its IP address using the AT+WIPADDR? command (**172.18.2.170** in this example).



2. Create a UDP socket on STA2 using the AT+SOPEN command (Port: 2002).

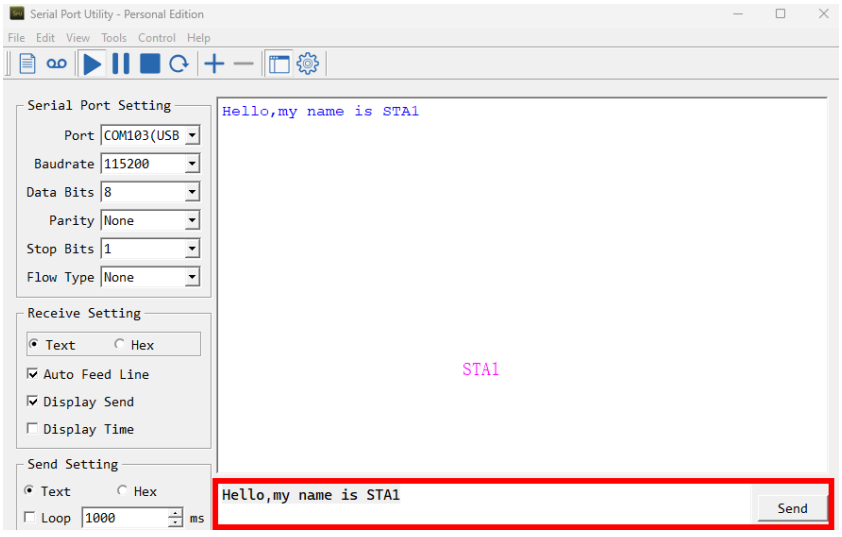


3. Refer to Section 3.2 to complete the DTU-mode parameter configuration for STA1 (UDP mode). Ensure STA1 is set to connect to the same HaLow AP.

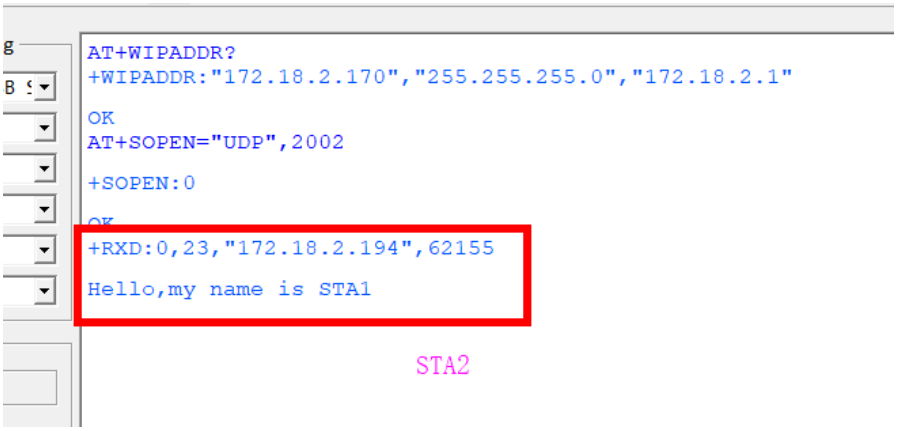


4. Power-cycle the module to apply the settings, and STA1 automatically connects to the target UDP device (STA2).

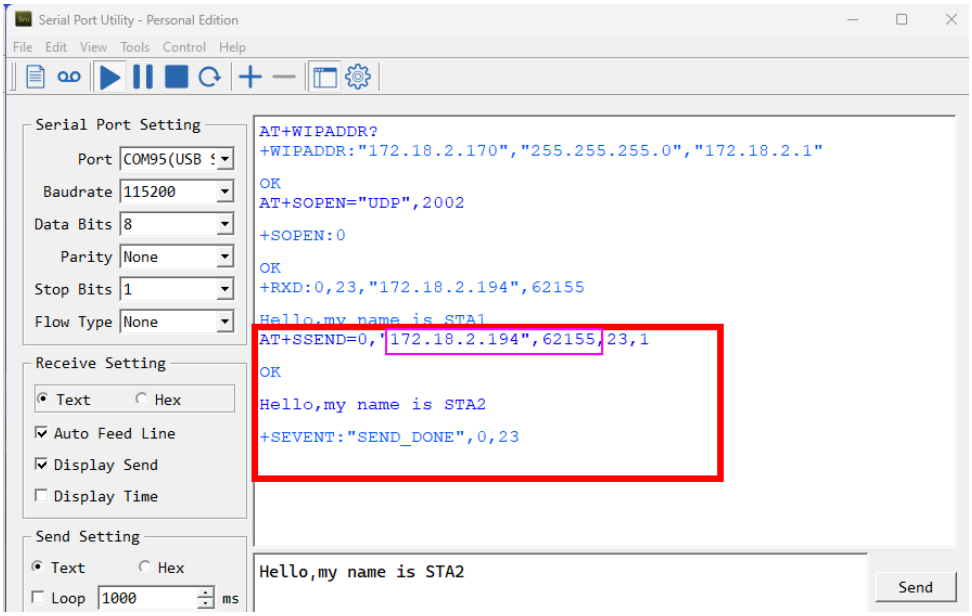
5. Send the string “Hello,my name is STA1” directly from STA1.



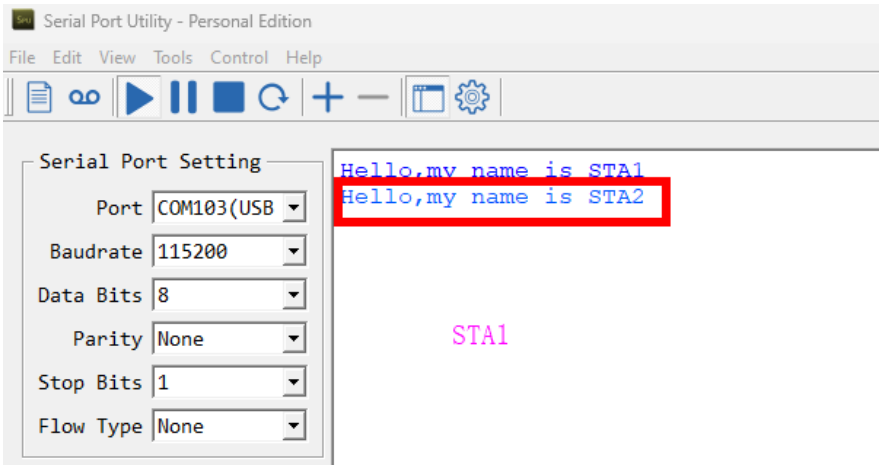
6. STA2 defaults to **Active Verbose Mode** for data reception and display if not configured.
   - Refer to Sections 2.5.7 & 2.5.8 for the mode configuration methods.
   - The +RXD message displays the IP address and port of STA1, which can be used by STA2 to send data to STA1.



7. To send the string “Hello,my name is STA2” from STA2 to STA1:
   - Send: AT+SSEND=<sock>,<remote_ip>,<port>,<data_size>,<done_event> (STA1’s IP and port are from the prior step)
   - Expected response: OK
   - Send: Hello,my name is STA2
   - Expected response: +SEVENT:"SEND_DONE",0,23



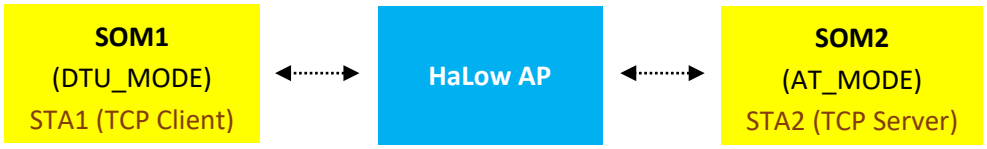
8. STA1 directly outputs the received data upon reception.



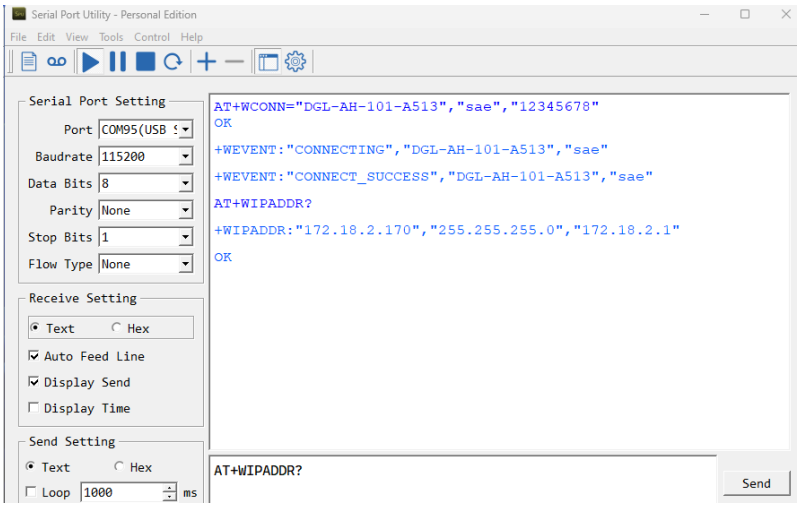
4.2.2 TCP Data Communication

Example: TCP Data Communication Between DTU-mode STA1 and AT-mode STA2.

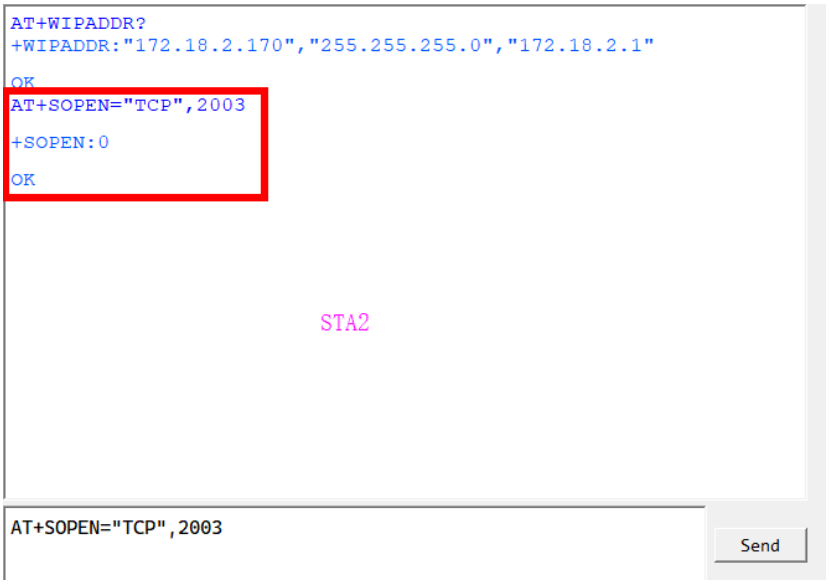
Configuration: STA1: Set to TCP mode; STA2: Create a TCP Server socket.



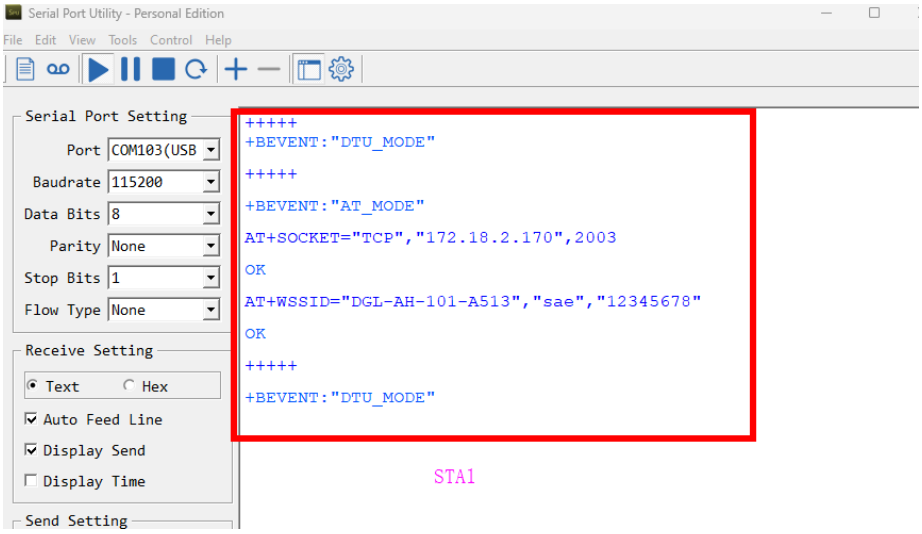
1. Connect STA2 to a HaLow AP, and query its IP address using the AT+WIPADDR? command (**172.18.2.170** in this example).



2. Create a TCP Server socket on STA2 using the AT+SOPEN command (Port: 2003).



3. Refer to Section 3.2 to complete the DTU-mode parameter configuration for STA1 (TCP mode). Ensure STA1 is set to connect to the same HaLow AP.



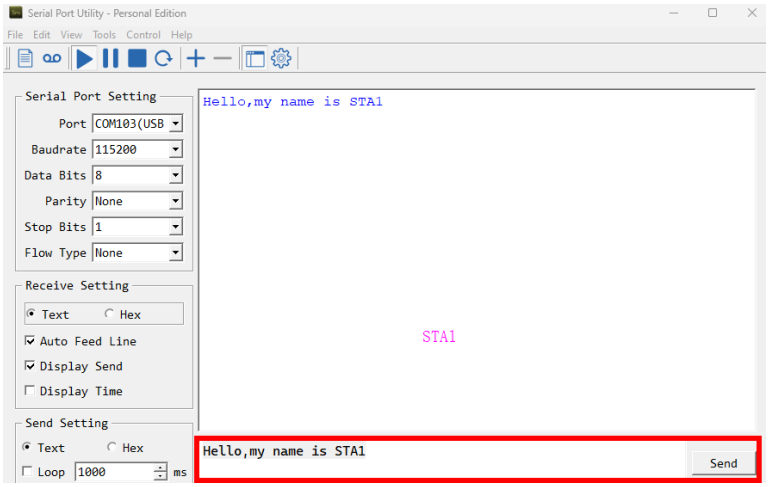
4. Power-cycle the module to apply the settings. STA1 will automatically connect to the TCP server (STA2).

5. When the connection establishes successfully, STA2 will display the connection information.

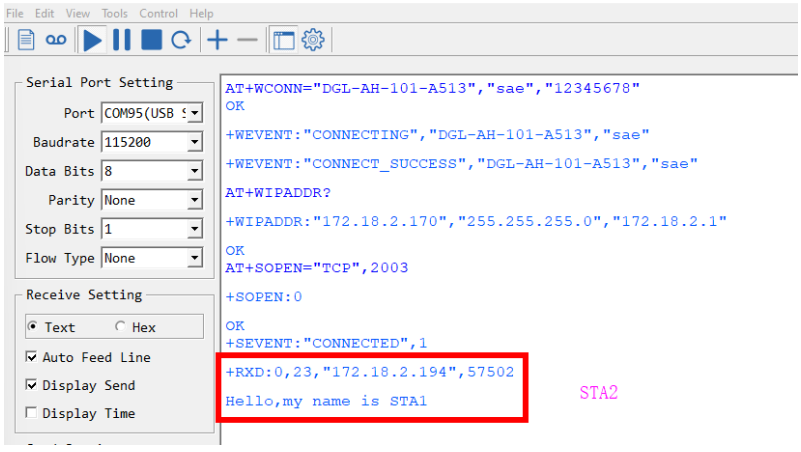
```
AT+WCONN="DGL-AH-101-A513","sae","12345678"
OK
+WEVENT:"CONNECTING","DGL-AH-101-A513","sae"
+WEVENT:"CONNECT_SUCCESS","DGL-AH-101-A513","sae"
AT+WIPADDR?
+WIPADDR:"172.18.2.170","255.255.255.0","172.18.2.1"
OK
AT+SOPEN="TCP",2003
+SOPEN:0
OK
+SEVENT:"CONNECTED",1
```

STA2

6. Send the string “Hello,my name is STA1” directly from STA1.

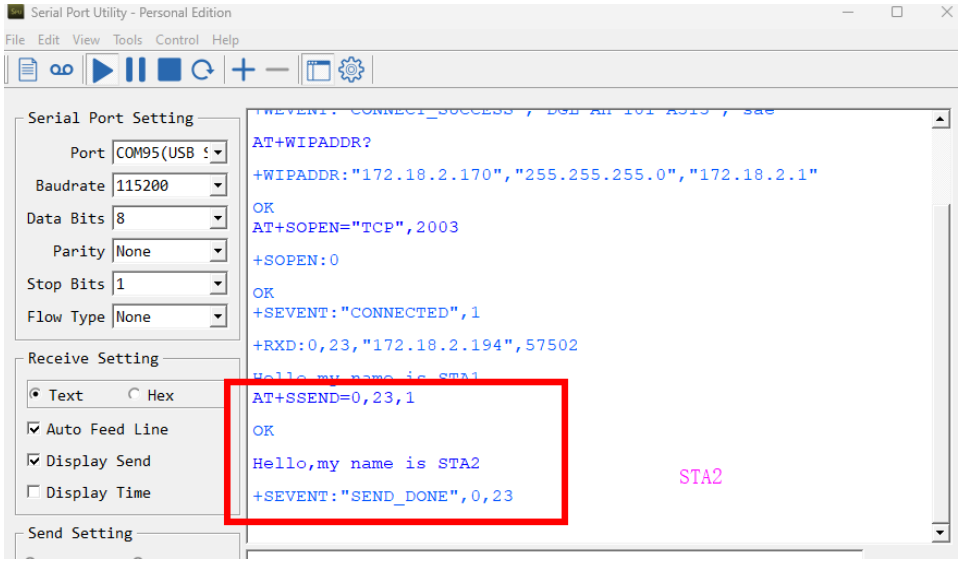


7. STA2 defaults to **Active Verbose Mode** for data reception and display if not configured.
   - Refer to Sections 2.5.7 & 2.5.8 for the mode configuration methods.
   - The +RXD message displays STA1’s IP address and port and the data sent from STA1.

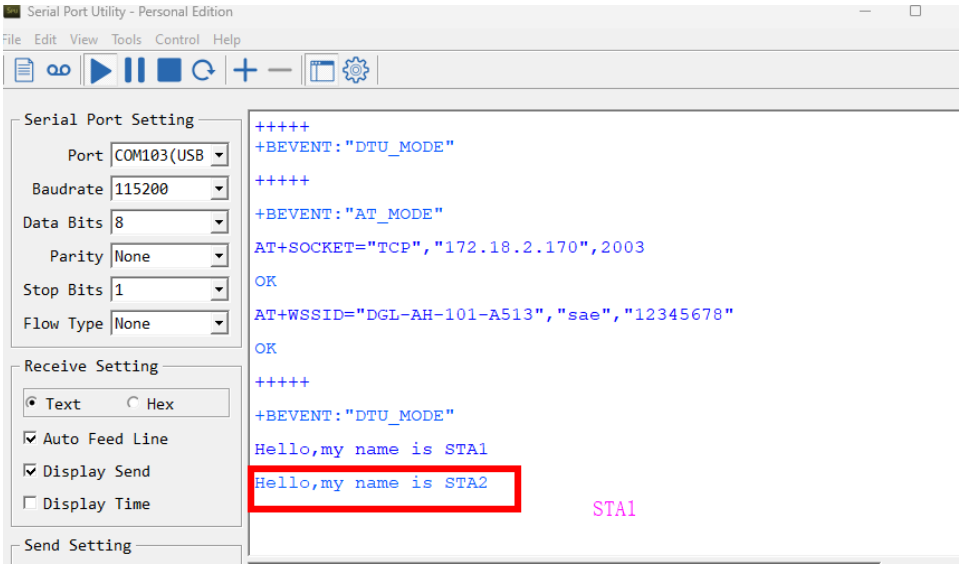


8. To send the string “Hello,my name is STA2” from STA2 to STA1:

   - Send: AT+SSEND=<sock>,<data_size>,<done_event>
   - Expected response: OK
   - Send: Hello,my name is STA2
   - Expected response: +SEVENT:"SEND_DONE",0,23



9. STA1 directly outputs the received data upon reception.



CHAPTER 5 IPERF TEST

iPerf is a network performance testing tool, and the iPerf version of Vantron's HaLow SOMs is **iPerf2**. Ensure that the iPerf versions between the AP and the module are consistent before conducting the iPerf test.

iPerf testing is only available when the HaLow module is in AT Command Mode. If the module is running as a TCP/UDP Server, you must restart it before switching its operation mode.

5.1 Installing Iperf

If iPerf is not installed on the HaLow AP side, refer to the AP's manual and install it first.

5.2 Testing Examples

Ensure that the module is connected to the HaLow AP before the iPerf test, otherwise the test cannot be properly executed.

5.2.1 IPERF_TCP_SERVER

Use the module as the Server and the HaLow AP as the Client.

1. Enter in the module's terminal the command: AT+WIPERF="TCP_SERVER",5001.

```
AT+WIPERF="TCP_SERVER",5001

OK
+WEVENT:Execute cmd on AP iperf -c 172.18.4.197 -p 5001 -i 1
```

   - An unsolicited event indicates that the HaLow AP needs to execute: iPerf -c 172.18.4.197 -p 5001 -i 1.

2. Execute the prompted command on the AP side.

```
root@VantronOS-1010:~# iperf -c 172.18.4.197 -p 5001 -i 1
------------------------------------------------------------
Client connecting to 172.18.4.197, TCP port 5001
TCP window size: 65.6 KByte (default)
------------------------------------------------------------
[  3] local 172.18.4.1 port 60570 connected with 172.18.4.197 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   356 KBytes  2.91 Mbits/sec
[  3]  1.0- 2.0 sec   357 KBytes  2.93 Mbits/sec
[  3]  2.0- 3.0 sec  92.7 KBytes   759 Kbits/sec
[  3]  3.0- 4.0 sec   135 KBytes  1.11 Mbits/sec
[  3]  4.0- 5.0 sec   221 KBytes  1.81 Mbits/sec
[  3]  5.0- 6.0 sec  85.5 KBytes   701 Kbits/sec
[  3]  6.0- 7.0 sec  21.4 KBytes   175 Kbits/sec
[  3]  7.0- 8.0 sec   221 KBytes  1.81 Mbits/sec
[  3]  8.0- 9.0 sec   200 KBytes  1.64 Mbits/sec
[  3]  9.0-10.0 sec   200 KBytes  1.64 Mbits/sec
[  3]  0.0-10.1 sec  1.84 MBytes  1.54 Mbits/sec
```

3. The module side will display the iPerf execution result:

```
+WEVENT:"IPERF_TCP_SERVER",Remote address "172.18.4.1",60570,local
address "172.18.4.197",5001,transferred 1 MBytes,duration 14655
ms,bandwidth 1055 kbps
```

Note: When the module acts as the iPerf Server, the device will not exit the server mode after the iPerf execution is completed. It will keep waiting for Client connections, and the AP side can initiate another test. If you need to change the iPerf mode of the module, send the AT+RST command to restart the device before reconfiguration.

5.2.2 IPERF_UDP_SERVER

Use the module as the Server and the HaLow AP as the Client.

1. Enter in the module's terminal the command: AT+WIPERF="UDP_SERVER",5001.

```
AT+WIPERF="UDP_SERVER",5001

OK
+WEVENT:Execute cmd on AP iperf -c 172.18.4.197 -p 5001 -i 1 -u -b 20M
```

   - An unsolicited event indicates that the HaLow AP needs to execute: iperf -c 172.18.4.197 -p 5001 -i 1 -u -b 20M

2. Execute the prompted command on the AP side.

```
root@VantronOS-1010:~# iperf -c 172.18.4.197 -p 5001 -i 1 -u -b 20M
------------------------------------------------------------
Client connecting to 172.18.4.197, UDP port 5001
Sending 1470 byte datagrams, IPG target: 560.76 us (kalman adjust)
UDP buffer size:  176 KByte (default)
------------------------------------------------------------
[  3] local 172.18.4.1 port 37581 connected with 172.18.4.197 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   251 KBytes  2.06 Mbits/sec
[  3]  1.0- 2.0 sec   208 KBytes  1.71 Mbits/sec
[  3]  2.0- 3.0 sec   297 KBytes  2.43 Mbits/sec
[  3]  3.0- 4.0 sec   233 KBytes  1.91 Mbits/sec
[  3]  4.0- 5.0 sec   237 KBytes  1.94 Mbits/sec
[  3]  5.0- 6.0 sec   240 KBytes  1.96 Mbits/sec
[  3]  6.0- 7.0 sec   257 KBytes  2.11 Mbits/sec
[  3]  7.0- 8.0 sec   287 KBytes  2.35 Mbits/sec
[  3]  8.0- 9.0 sec   240 KBytes  1.96 Mbits/sec
[  3]  0.0-10.1 sec  2.42 MBytes  2.02 Mbits/sec
[  3] Sent 1727 datagrams
```

3. The module side will display the iPerf execution result:

```
+WEVENT:"IPERF_UDP_SERVER",Remote address "172.18.4.1",37581,local
address "0.0.0.0",5001,transferred 2 MBytes,duration 10194 ms,bandwidth
1972 kbps
```

Note: When the module acts as the iPerf Server, the device will not exit the server mode after the iPerf execution is completed. It will keep waiting for Client connections, and the AP side can initiate another test. If you need to change the iPerf mode of the module, send the AT+RST command to restart the device before reconfiguration.

5.2.3 IPERF_TCP_CLIENT

Use the module as the Client and the HaLow AP as the Server.

1. Enter in the HaLow AP's terminal the iPerf -s -i 1 command.

```
root@VantronOS-4040:/# iperf -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
```

2. Execute the command AT+WIPERF="TCP_CLIENT",5001,"172.18.4.1",10 on the module.

   - **172.18.4.1** is the gateway IP address, which can be viewed via the AT+WIPADDR? command.

```
AT+WIPADDR?

+WIPADDR:"172.18.4.197","255.255.255.0","172.18.4.1"
OK
```

3. After executing above command, the AP side will display the following information.

```
[  4] local 172.18.4.1 port 5001 connected with 172.18.4.197 port 56691
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 1.0 sec  1.45 KBytes  11.9 Kbits/sec
[  4]  1.0- 2.0 sec  1.43 KBytes  11.7 Kbits/sec
[  4]  2.0- 3.0 sec   135 KBytes  1.11 Mbits/sec
[  4]  3.0- 4.0 sec  32.8 KBytes   269 Kbits/sec
[  4]  4.0- 5.0 sec  31.4 KBytes   257 Kbits/sec
[  4]  5.0- 6.0 sec  7.13 KBytes  58.4 Kbits/sec
[  4]  6.0- 7.0 sec   123 KBytes  1.00 Mbits/sec
[  4]  7.0- 8.0 sec  7.13 KBytes  58.4 Kbits/sec
[  4]  8.0- 9.0 sec  71.3 KBytes   584 Kbits/sec
[  4]  9.0-10.0 sec  0.00 Bytes  0.00 bits/sec
[  4]  0.0-10.5 sec   419 KBytes   326 Kbits/sec
```

4. The module side will display the iPerf execution result:

```
+WEVENT:"IPERF_TCP_CLIENT",Remote address "172.18.4.1",5001,local
address "172.18.4.197",56691,transferred 429 KBytes,duration 10503
ms,bandwidth 326 kbps
```

Note: When the module acts as the iPerf Client, the device will exit the iPerf mode after the iPerf test is completed. No reset operation is required, and the module can perform other iPerf operations directly.

5.2.4 IPERF_UDP_CLIENT

Use the module as the Client and the HaLow AP as the Server.

1. Enter in the HaLow AP's terminal the iPerf -s -i 1 -u -b 20M command.

```
root@VantronOS-4040:/# iperf -s -i 1 -u -b 20M
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  176 KByte (default)
------------------------------------------------------------
```

2. Execute the command AT+WIPERF="UDP_CLIENT",5001,"172.18.4.1",10 on the module.

   - **172.18.4.1** is the gateway IP address, which can be viewed via the AT+WIPADDR? command.

```
AT+WIPADDR?

+WIPADDR:"172.18.4.197","255.255.255.0","172.18.4.1"
OK
```

3. After executing above command, the AP side will display the following information.

```
[  3] local 172.18.4.1 port 5001 connected with 172.18.4.197 port 5013
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec   279 KBytes  2.29 Mbits/sec  10.724 ms    0/  196 (0%)
[  3]  1.0- 2.0 sec   279 KBytes  2.29 Mbits/sec  10.000 ms    0/  196 (0%)
[  3]  2.0- 3.0 sec   278 KBytes  2.28 Mbits/sec   7.774 ms    0/  195 (0%)
[  3]  3.0- 4.0 sec   279 KBytes  2.29 Mbits/sec  10.095 ms    0/  196 (0%)
[  3]  4.0- 5.0 sec   289 KBytes  2.37 Mbits/sec   7.023 ms    0/  203 (0%)
[  3]  5.0- 6.0 sec   279 KBytes  2.29 Mbits/sec   8.712 ms    0/  196 (0%)
[  3]  6.0- 7.0 sec   278 KBytes  2.28 Mbits/sec   8.399 ms    0/  195 (0%)
[  3]  7.0- 8.0 sec   278 KBytes  2.28 Mbits/sec   9.090 ms    0/  195 (0%)
[  3]  8.0- 9.0 sec   297 KBytes  2.43 Mbits/sec   7.909 ms    0/  208 (0%)
[  3]  9.0-10.0 sec   297 KBytes  2.43 Mbits/sec   9.715 ms    0/  208 (0%)
[  3]  0.0-10.1 sec  2.79 MBytes  2.32 Mbits/sec   8.779 ms    0/ 2007 (0%)
```

4. The module side will display the iPerf execution result:

```
+WEVENT:"IPERF_UDP_CLIENT",Remote address "172.18.4.1",5001,local address
"0.0.0.0",5013,transferred 2 MBytes,duration 10089 ms,bandwidth 2323 kbps
```

Note: When the module acts as the iPerf Client, the device will exit the iPerf mode after the iPerf test is completed. No reset operation is required, and the module can perform other iPerf operations directly.