twareLAB

4CH S2E Module

User Manual Ver1.0

twareLAB

Contents

Table of Contents

Table of Contents

1 INTRODUCTION

1.1 KEY FEATURES

- 4 Port Serial-to-Ethernet
- Support DHCP IP Acquisition
- Support DNS Query
- Support NTP Time Query
- Support Time Zone Setting
- TCP/UDP Data Communication
- Ethernet Data Packing Option
- Support Up to 3Mbps UART Baud Rate

1.2 PRODUCT SPECIFICATIONS

	Specification
MCU	STM32F405RGT6 (RAM: 192Kbyte, FLASH: 1Mbyte)
LAN	W5500 (10/100Mbps Ethernet)
UART	4 Port (3.3V TTL Level)
Console Port	Supported
Dimension	TW100MJ: 48.26(W)x61.4(H)x22(D), TW100XR: 48.26(W)x58.00(H)x15(D)
Connector	2.54mm Pitch Pin Header. J5: 2x14, J6: 1x14
Input Power	DC 3.3V
Power Dissipation	Typical: 100mA, Peak: 150mA
Temperature	Operation: 0°C ~ +70°C, Storage: -40°C ~ +85°C
Humidity	10 ~ 80%

Table 1 Product Specifications

2 GETTING STARTED

2.1 HARDWARE SETTING

The procedure for testing the TW100xx EVB follows the below sequence.

① Connect the USB mini cable included in the EVB Box to the J7 USB MINI SOCKET.
② Connect the USB Type A terminal to the computer.
③ Change the power switch (SW4) to the [ON] part.
④ When the LD1 LED lights up, it means that the power is applied normally.
⑤ In Windows, the COM port number can be shown in “Device Manager” window. In case of Linux, the COM port can be listed by running ls /dev/ command on the shell.
⑥ Run the Serial Terminal Program and open the Com port. **The default Baud Rate is 2Mbps.**
⑦ If no message is displayed, press SW1 (RESET SW) to hardware reset the module.
⑧ If the message shown in the figure below, the hardware setting has been completed.

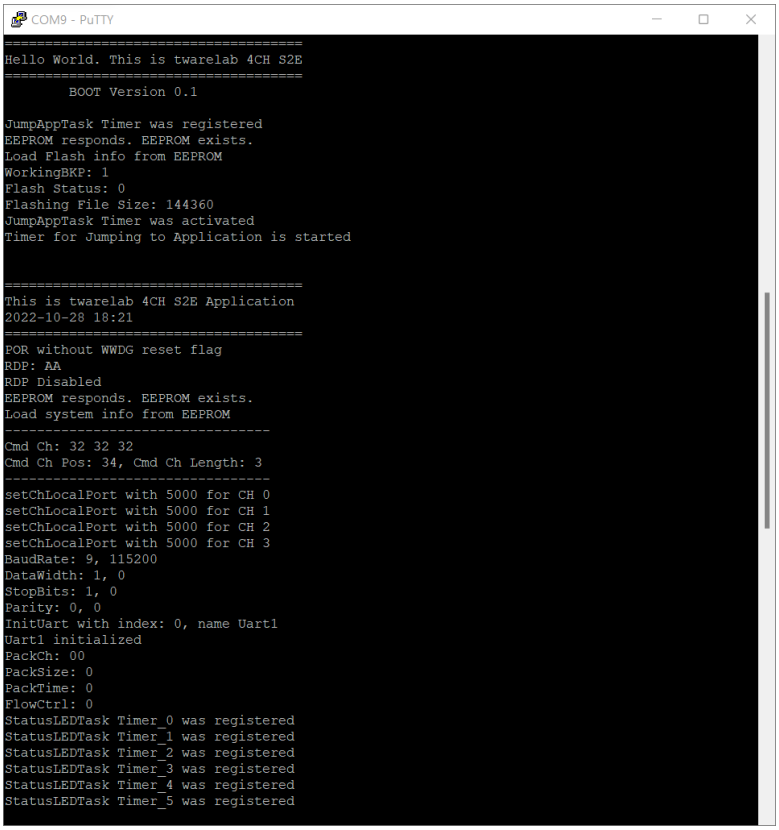

Figure 1 Debug Message Screen at Boot

2.1.1 Debug Port

entry	Value
Baud Rate	2Mbps
Data Bit	8
Stop Bit	1
Parity	None
Flow Ctrl	None

Table 2Settings for Debug UART Ports

2.2 Factory Default Value

classification	entry		Value
General Info	DHCP		Disable (Static IP)
	Debug		Disable
	Local IP Address		192.168.0.100
	Local Subnet		255.255.255.0
	Local Gateway		192.168.0.1
	DNS Server IP		168.126.63.1
Channel Info	Mode		Server Mode
	Connection Status		Disconnected
	DNS		Disable
	UDP		Disable
	Remote IP		0.0.0.0
	Local Port		0
	Remote Port		0
	Domain Name		NULL
	UART	Baud Rate	115200
		Data bit	8
		Stop bit	1
		Parity	None
	Packing Condition	Char	Disable (0x00)
		Size	Disable (0)
		Time	Disable (0)
	Inactivity Time		Disable (0)

Table 3 Factory reset values

2.3 RESET

There are two ways to reset a module.

Hardware Reset, which controls the /RESET pin connected to the NRST pin of the MCU, and Software Reset, which controls the SW_INPUT pin connected to the GPIO of the MCU.

Software Reset is divided into “General Software Reset” and “Factory Reset” according to the time it takes to keep the SW_INPUT LOW.

2.3.1 Hardware Reset

It is the same as the specification of the NRST pin of MCU.

2.3.2 Software Reset

Keeping the SW_Input pin at least 100ms LOW will enter the Software Reset Enable state, and if you input the SW_Input pin to HIGH, the module will be Software Reset.

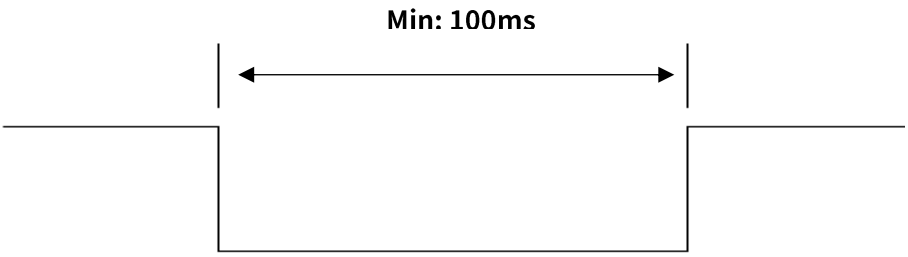

Figure 2 Software Reset Signal Timing Diagram

2.3.3 Factory Reset

If the SW_Input pin is kept at 5 seconds LOW, the “Factory Reset Enable” state is entered, and then you input the SW_Input pin to HIGH, the module changes all setting values to their initial factory settings.

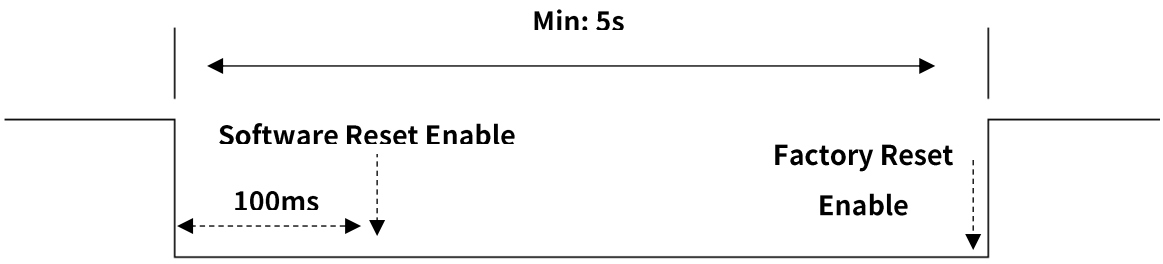

Figure 3 Factory Reset Signal Timing Diagram

2.4 COMMAND MODE

HW_TRIGGER pin can be controlled to turn AT Command Mode on/off.

When the HW_TRIGGER pin go to LOW, it allows the module to be controlled via AT Command.

UART1 port only accepts AT Command. When the HW_TRIGGER pin is HIGH, UART1 is used for data communication, and when the HW_TRIGGER pin is LOW, UART1 is used for AT commands channel.

For details, see the section AT Commands.

3 CONFIGURATION TOOL

twareLAB Standard Configuration Tool is a PC application used to set the configuration data on modules. There are two versions, one for Windows and one for Linux, and they are available as freeware.

Figure 4below shows the initial screen displayed after running the Configuration Tool.

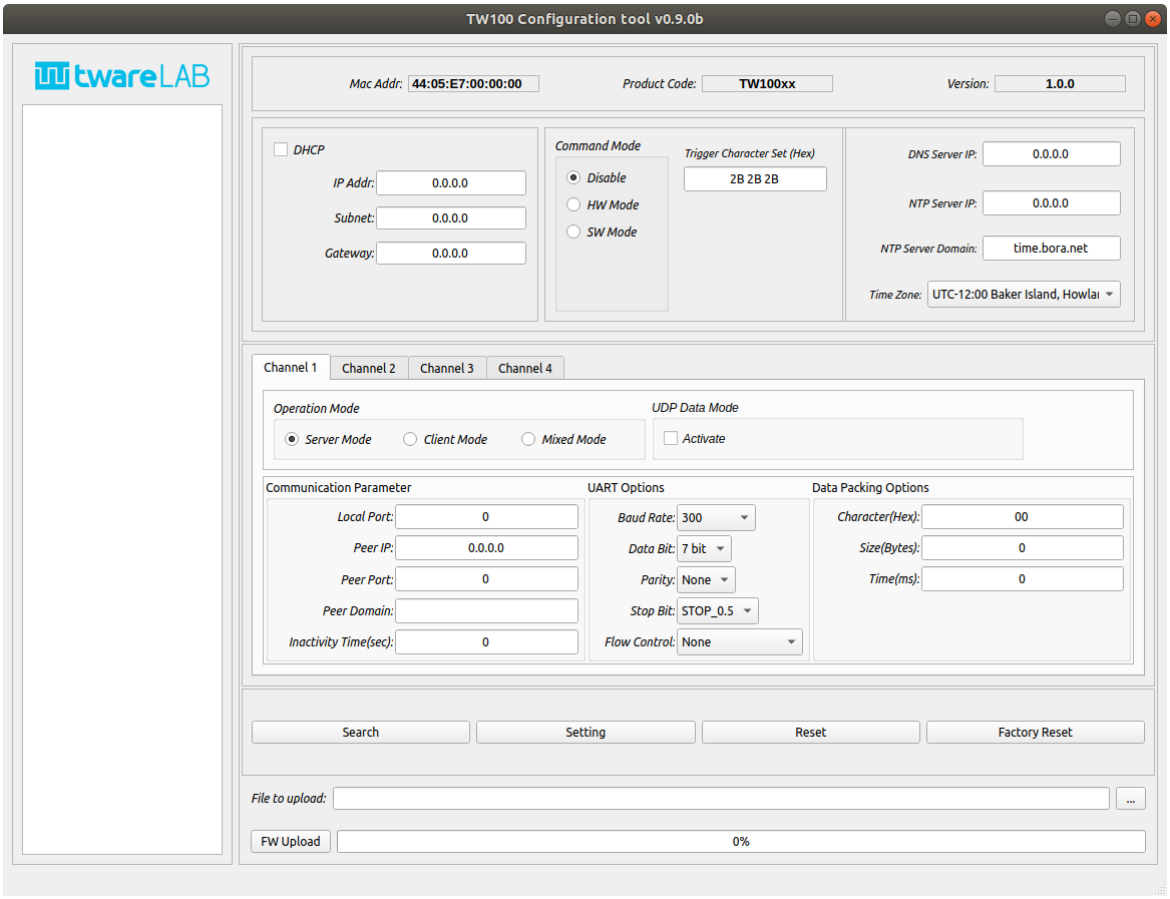

Figure 4 twareLAB Configuration Tool splash screen

Device search, configuration, reset, Factory reset, and firmware update functions can be performed, see "TW100XX Configuration Tool Manual" for detailed instructions.

The figure below shows the screen after Device search.

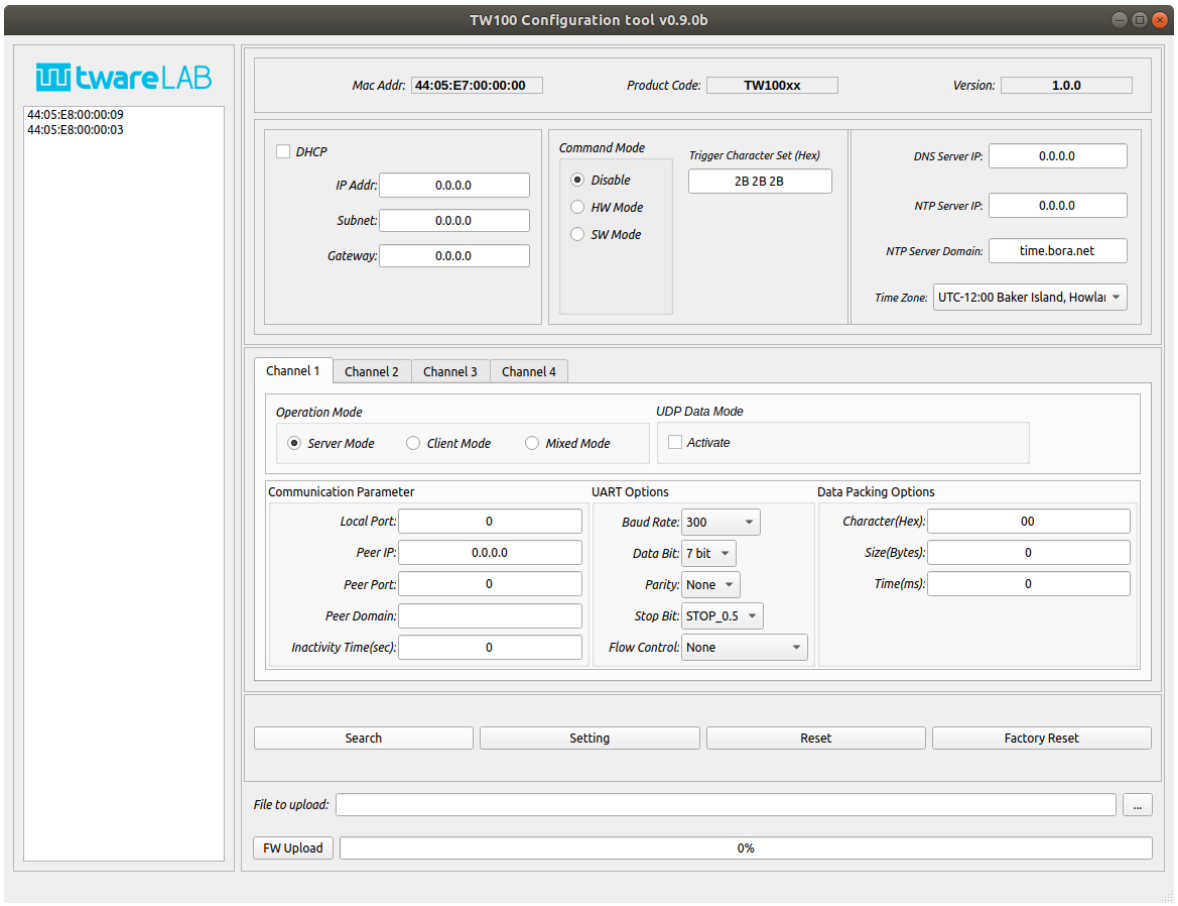

Figure 5 Device Search Results Screen

The following is the firmware update screen.

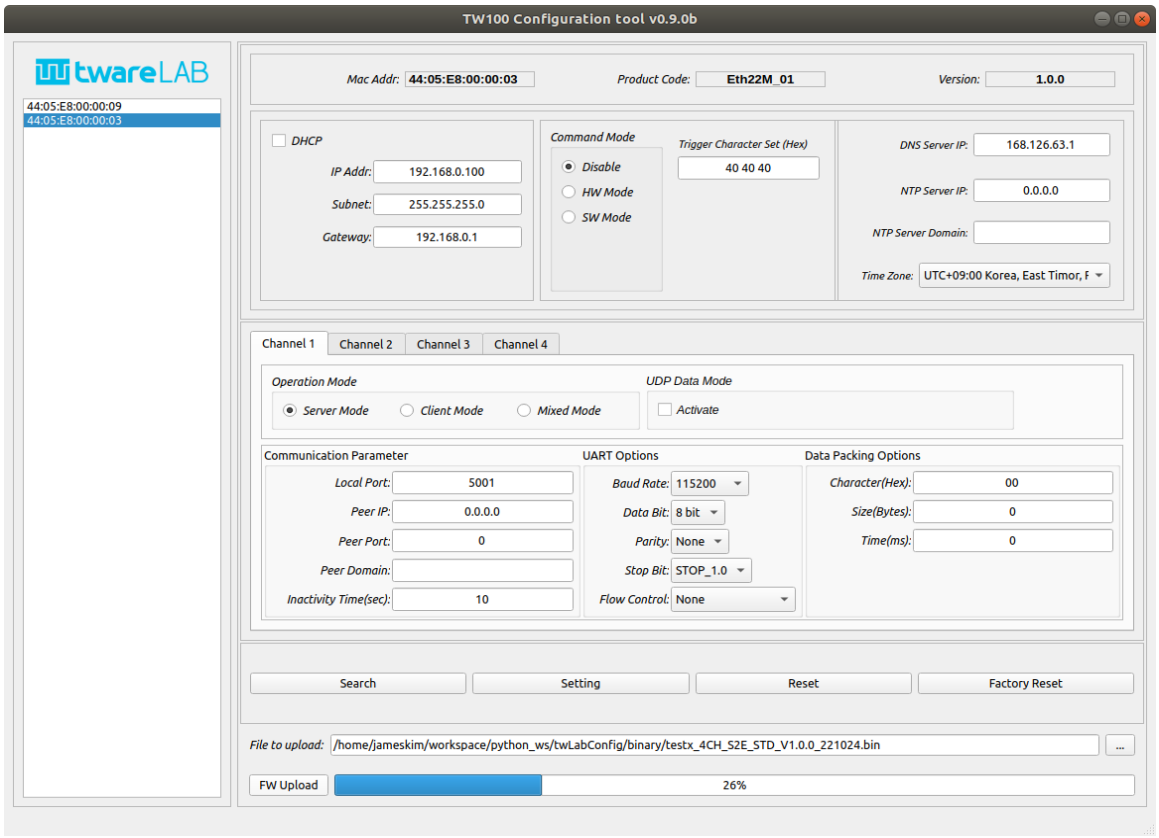

Figure 6 Firmware Upload Screen

If it is performed normally, the following message window will appear.

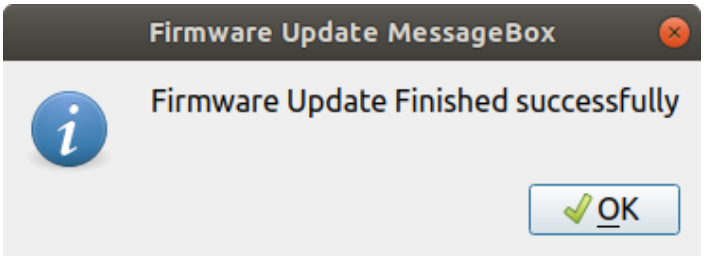

Figure 7Box displayed after firmware upload is complete

4 AT COMMANDS

Attention) All AT commands must always end with '\r\n' at the end.

Function	Command Syntax
Command mode check	AT\r\n
Reboot the module	AT+REBOOT\r\n
Check PRODUCT ID	AT+PRODUCTID?\r\n
Check the firmware version	AT+VER?\r\n
MAC Address Verification	AT+MAC?\r\n
Check your registration status	AT+REG?\r\n
Request registration	AT+REG=<option>\r\n
Check network information	AT+DNETINFO?\r\n
Network Information Settings	AT+DNETINFO=<ip mode>,<local ip>,<subnet>,<gateway>\r\n
Check the DNS server address	AT+DNSIP?\r\n
Setting the DNS Server Address	AT+DNSIP=<dns server IP>\r\n
DNS Query request	AT+DNSQUERY?<domain name>\r\n
Check NTP Server Address	AT+NTP?\r\n
Setting NTP server address	AT+NTP=<ntp server url>\r\n
Check the current time	AT+TIME?\r\n
Check Time Zone	AT+TZONE?\r\n
Time Zone Settings	AT+TZONE=<time zone>\r\n
Check UART Information	AT+UART?\r\n
Setting UART Information	AT+UART=<uart num>,<uart*1* setting>,…<uart*n* setting>\r\n
Check Peer information	AT+PEERINFO?\r\n
Setting Peer Information	AT+PEERINFO=<peer num>,<peer*1* setting>,…<peer*n* setting>\r\n
FWUP Start	AT+FWUPSTART=<file size>\r\n
FWUP Data	AT+FWUPDATA=<data>\r\n
FWUP Finish	AT+FWUPFINISH\r\n
SWI NPUT check	AT+SWINPUT?\r\n
SWINPUT Settings	AT+SWINPUT=<option>\r\n
CRC16 Usage Verification	AT+USECRC16?\r\n
Enabling CRC16	AT+USECRC16=<option>\r\n

Table 4 AT Command List

4.1 CHECK COMMAND MODE

4.1.1 command

```
AT\r\n
```

4.1.2 response

```
OK\r\n
```

4.1.3 Example

```
AT\r\n
OK\r\n
```

4.2 REBOOT THE MODULE

4.2.1 command

```
AT+REBOOT\r\n
```

4.2.2 response

```
+REBOOT\r\n
OK\r\n
```

4.2.3 Example

```
AT+REBOOT\r\n
+REBOOT\r\n
OK\r\n
```

4.3 CHECK PRODUCT ID

4.3.1 command

```
AT+PRODUCTID? \r\n
```

4.3.2 response

```
+PRODUCTID=<productid>\r\n
OK\r\n
```

4.3.3 Example

```
AT+PRODUCTID? \r\n
+PRODUCTID=0x0101\r\n
OK\r\n
AT+VER? \r\n
```

4.4 CHECK THE FIRMWARE VERSION

4.4.1 command

```
AT+VER? \r\n
```

4.4.2 response

```
+VER=<version>\r\n
OK\r\n
```

4.4.3 Example

```
AT+VER? \r\n
+VER=1.0.9\r\n
OK\r\n
```

4.5 MAC Address Verification

4.5.1 command

```
AT+MAC? \r\n
```

4.5.2 response

```
+MAC=<mac addr>\r\n
OK\r\n
```

4.5.3 Example

```
AT+MAC? \r\n
+MAC=44:05:E8:12:34:56\r\n
OK\r\n
```

4.6 Check your registration status

4.6.1 command

```
AT+REG? \r\n
```

4.6.2 response

```
+REG=<status>\r\n
OK\r\n
```

4.6.3 Example

```
AT+REG? \r\n
+REG=0\r\n
OK\r\n
```

4.7 Request registration

4.7.1 command

```
AT+REG=<option>\r\n
```

4.7.2 response

```
+REG=<option>\r\n
OK\r\n
```

4.7.3 Example

```
AT+REG=1\r\n
+REG=1\r\n
OK\r\n
```

4.8 Requesting Network Information

4.8.1 command

```
AT+ DNETINFO?\r\n
```

4.8.2 response

```
+DNETINFO=<ip mode>,<local ip>,<subnet>,<gateway>\r\n
OK\r\n
```

4.8.3 Example

```
AT+DNETINFO? \r\n
+DNETINFO=0,192.168.0.100,255.255.255.0,192.168.0.1\r\n
OK\r\n
```

4.9 Network Information Settings

4.9.1 command

```
AT+DNETINFO=<ip mode>,<local ip>,<subnet>,<gateway>\r\n
```

4.9.2 response

```
+DNETINFO=<ip mode>,<local ip>,<subnet>,<gateway>\r\n
OK\r\n
```

4.9.3 Example

```
AT+DNETINFO=0, 192.168.0.101, 255.255.255.0, 192.168.0.1\r\n
+DNETINFO=0, 192.168.0.101, 255.255.255.0, 192.168.0.1 \r\n
OK\r\n
```

4.10 Check the DNS server address

4.10.1 command

```
AT+DNS? \r\n
```

4.10.2 response

```
+DNS=<dns server ip>\r\n
OK\r\n
```

4.10.3 Example

```
AT+DNS? \r\n
+DNS=168.126.63.1\r\n
OK\r\n
```

4.11 Setting the DNS Server Address

4.11.1 command

```
AT+DNS=<dns server ip>\r\n
```

4.11.2 response

```
+DNS=<dns server ip>\r\n
OK\r\n
```

4.11.3 Example

```
AT+DNS=168.126.63.1\r\n
+DNS=168.126.63.1\r\n
OK\r\n
```

4.12 CHECK THE NTP SERVER ADDRESS

4.12.1 command

```
AT+NTP? \r\n
```

4.12.2 response

```
+NTP=<ntp server domain>\r\n
OK\r\n
```

4.12.3 Example

```
AT+NTP? \r\n
+NTP=time.bora.net\r\n
OK\r\n
```

4.13 SETTING NTP SERVER ADDRESS

4.13.1 command

```
AT+NTP=<ntp server domain>\r\n
```

4.13.2 response

```
+NTP=<ntp server domain>\r\n
OK\r\n
```

4.13.3 Example

```
AT+NTP=time.bora.net\r\n
+NTP=time.bora.net\r\n
OK\r\n
```

4.14 CHECK THE CURRENT TIME

4.14.1 command

```
AT+TIME? \r\n
```

4.14.2 response

```
+TIME=YYYY/MM/DD/hh:mm:ss, <Day of Week>, <Time Zone>\r\n
OK\r\n
```

4.14.3 Example

```
AT+TIME? \r\n
+TIME=2022/08/03/02:44:33, WED, <?>\r\n
OK\r\n
```

4.15 TIME ZONE SETTINGS

4.15.1 command

```
AT+TZONE=<Time Zone>\r\n
```

4.15.2 response

```
+TZONE=<Time Zone>\r\n
OK\r\n
```

4.15.3 Example

```
AT+TZONE=+1\r\n
+TZONE=+1\r\n
OK\r\n
```

4.16 CHECK UART INFORMATION

4.16.1 command

```
AT+UART? \r\n
```

4.16.2 response

```
+UART=<uart num>, <uart 1 setting >,...,<uart n setting>\r\n
OK\r\n
```

4.16.2.1 Uart setting

Individual options use numeric characters between 0 ~ 15, and connects options with the Dash('-') character.

The order of the options is as follows.

<BaudRate>-<Databit>-<Stopbit>-<Parity>-<FlowCtrl>-<PackCH>-<PackSize>-<PackTime>-<InactivitiTime>

4.16.2.2 BaudRate

BaudRate(bps)	Value
300	0
600	1
1,200	2
2,400	3
4,800	4
9,600	5
19,200	6

38,400	7
57,600	8
115,200	9
230,400	10
460,800	11
921,600	12
1,000,000	13
2,000,000	14
3,000,000	15

Table 5 BaudRate Index

4.16.2.3 DataBit

Databit	Value
7 (Not Support yet)	0
8 (Default)	**1**

Table 6 DataBit Index

4.16.2.4 StopBit

Stopbit	Value
0.5 (Not support yet)	0
1 (Default)	**1**
1.5 (Not support yet)	2
2 (Not support yet)	3

Table 7 StopBit Index

4.16.2.5 Parity

Parity	Value
None	0
ODD	1
EVEN	2

Table 8 Parity Option Index

4.16.2.6 FlowCtrl

FlowCtrl	Value
None	0
XON/XOFF	1
RTS/CTS	2
RTS ONLY	3
Reverse RTS ONLY	4

Table 9 FlowCtrl Option Index

4.16.2.7 PackCH

Represents a character as a Hex value.

Space(= 20), '+' (= 32)

4.16.2.8 PackSize

A value between 0 ~ 1000 can be specified as a string.

4.16.2.9 PackTime

A value between 0 ~ 1000 can be specified as a string.

4.16.2.10InactivityTime

A value between 0 ~ 3600 can be specified as a string.

4.16.3 Example

```
AT+UART? \r\n
+UART=1,9-1-1-0-2-00-0-0-10\r\n
OK\r\n
```

In the example above, one UART is set, Baudrate is 9(= 115200), Databit is 1(= 8bit), Stopbit is 1(= 1 bit), Parity is 0(= None), FlowCtrl is 2 (= RTS/CTS), PackCH is 00 (= Disable), and PackSize is 0(= Disable), PackTime is set to 0(= Disable), and Inactivity Time is set to 10(= 10 seconds).

4.17 SETTING UART INFORMATION

4.17.1 command

AT+UART=<uart num>, <uart *1* setting **>,...,<uart** n **setting>\r\n**

4.17.2 response

+UART=<uart num>, <uart *1* setting **>,...,<uart** n **setting>\r\n**
OK\r\n

4.17.3 Example

AT+UART=1,9-1-1-0-2-00-0-0-10\r\n
+UART=1,9-1-1-0-2-00-0-0-10\r\n
OK\r\n

4.18 CHECK PEER INFORMATION

4.18.1 command

AT+PEERINFO? \r\n

4.18.2 response

+PEERINFO=<peer **num>,<peer***1*** setting **>,..., <**peer n **setting>**\r\n
OK\r\n

4.18.2.1 Peer Setting

Individual options use numeric characters between 0 ~ 9, and connects options with the Dash('-') character.

<Operation Mode>-<Connection Status>-<bDNS>-<bUDP>-<Local Port>-<Remote IP>-<Remote Port**>-<Remote Domain>**

4.18.2.2 Operation Mode

Operation Mode	Value

Server Mode	0
Client Mode	1

Table 10 Operation Mode Index

4.18.2.3 Connection Status

Connection Status	Value
Disconnected	0
Connected	1

Table 11 Connection Status Index

4.18.2.4 bDNS

DNS	Value
Disable	0
Enable	1

Table 12 DNS Option Index

4.18.2.5 bUDP

UDP	Value
TCP mode	0
UDP mode	1

Table 13 UDP Option Index

4.18.2.6 Local Port

Values between 0 ~ 65535 are expressed as strings.

4.18.2.7 Remote IP

xxx.xxx.xxx.xxx as a string.

4.18.2.8 Remote Port

Values between 0 ~ 65535 are expressed as strings.

4.18.2.9 Remote Domain

It is expressed as a string.

Specifies the string Null or domain name.

4.18.3 Example

```
AT+PEERINFO? \r\n
+ PEERINFO=1,0-0-1-0-5000-192.168.0.110-6000-\r\n
OK\r\n
```

In the above example, This means that one socket to communicate with the peer system is enabled, and for communication with the peer system, Operation Mode is 0(= Server Mode), the current Connection Status is 0(= Disconnected), DNS function is 1(= Enable), UDP is 0 (= TCP mode), Local Port is 5000, the Remote IP is set to 192.168.0.110, the Remote Port is set to 6000, and the Remote Domain is not specified.

4.19 Setting Peer Information

4.19.1 command

```
AT+PEERINFO=<peer num>, <peer 1 setting >,...,<peer n setting>\r\n
```

4.19.2 response

```
+PEERINFO=<peer num>,<peer1 setting >,...,<peer n setting>\r\n
OK\r\n
```

4.19.3 Example

```
AT+PEERINFO=1,0-0-1-0-5000-192.168.0.110-6000-\r\n
+PEERINFO=1,0-0-1-0-5000-192.168.0.110-6000-\r\n
OK\r\n
```

※ Note) Even if there is no Remote domain, '-' must be entered at the end.

5 Flow Control Mode

5.1 RTS/CTS Mode

RTS / CTS Flow Control is a simple flow control mechanism that is part of the RS232 standard. It makes use of two further pins on the RS232 connector, RTS (Request to Send) and CTS (Clear to Send). These two lines allow the receiver and the transmitter to alert each other to their state.

When the receive buffer of Uart becomes Full, the RTS pin goes to HIGH, and when the receive buffer is released by more than a certain size, the RTS pin is output to LOW.

In addition, if the CTS pin is HIGH, data transmission (the TX pin of the UART) stops, and when the CTS pin drops LOW, data transmission can start.

5.2 Xon/Xoff Mode

V1.0.0 does not support it.

5.3 RTS Only Mode

When connecting a UART to the R S485 Transceiver, the RTS pin is used for TX control of the RS485 Transceiver. You can use this mode if TX Enable is Active LOW.

The RTS signal drops to **LOW** before **loading** data into the TXD and switches to **HIGH after transmission is complete**.

Figure 8RTS Signal Timing Diagram in RTS Only Mode

5.4 Reverse RTS Only Mode

*) V1.0.0 does not support it.

When connecting a UART to the R S485 Transceiver, the RTS pin is used for TX control of the RS485 Transceiver. You can use this mode when TX Enable is Active HIGH.

The RTS signal goes up to **HIGH before** loading data into the TXD and switches to **LOW after transmission is complete**.

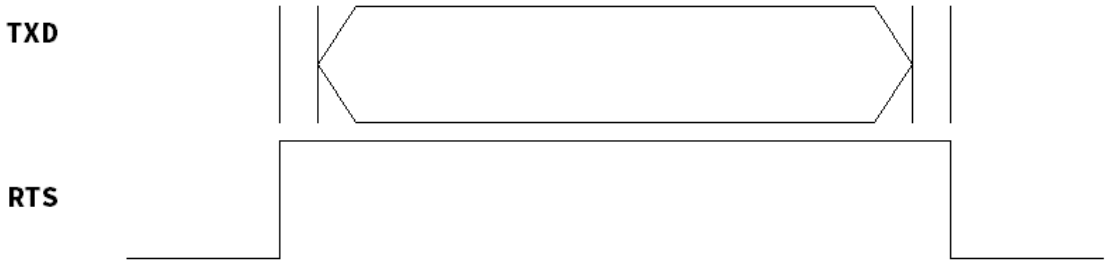

Figure 9RTS Signal Timing Diagram in Reverse RTS Only Mode

6 OPERATION GUIDE

TCP and UDP settings are required for data communication, support various mode options as TCP Server, TCP Client, TCP Mixed, and UDP mode. In addition, the Data packing condition option can set the delimiter options of transmission data.

6.1 TCP SERVER MODE

TCP Server mode means that the TW100XX module acts as a server during the TCP connection establishment process for serial data communication, and waits for a TCP connection from any peer on the specified local port.

Performs an operation to send serial data received over a connected TCP connection and to send data received over a TCP connection to the serial interface.

Therefore, for TCP server mode operation, the operation mode must be set to "Server Mode" using AT Command or Configuration tool, and the network settings such as Local IP, Subnet, Gateway, and Local Port must be set normally.

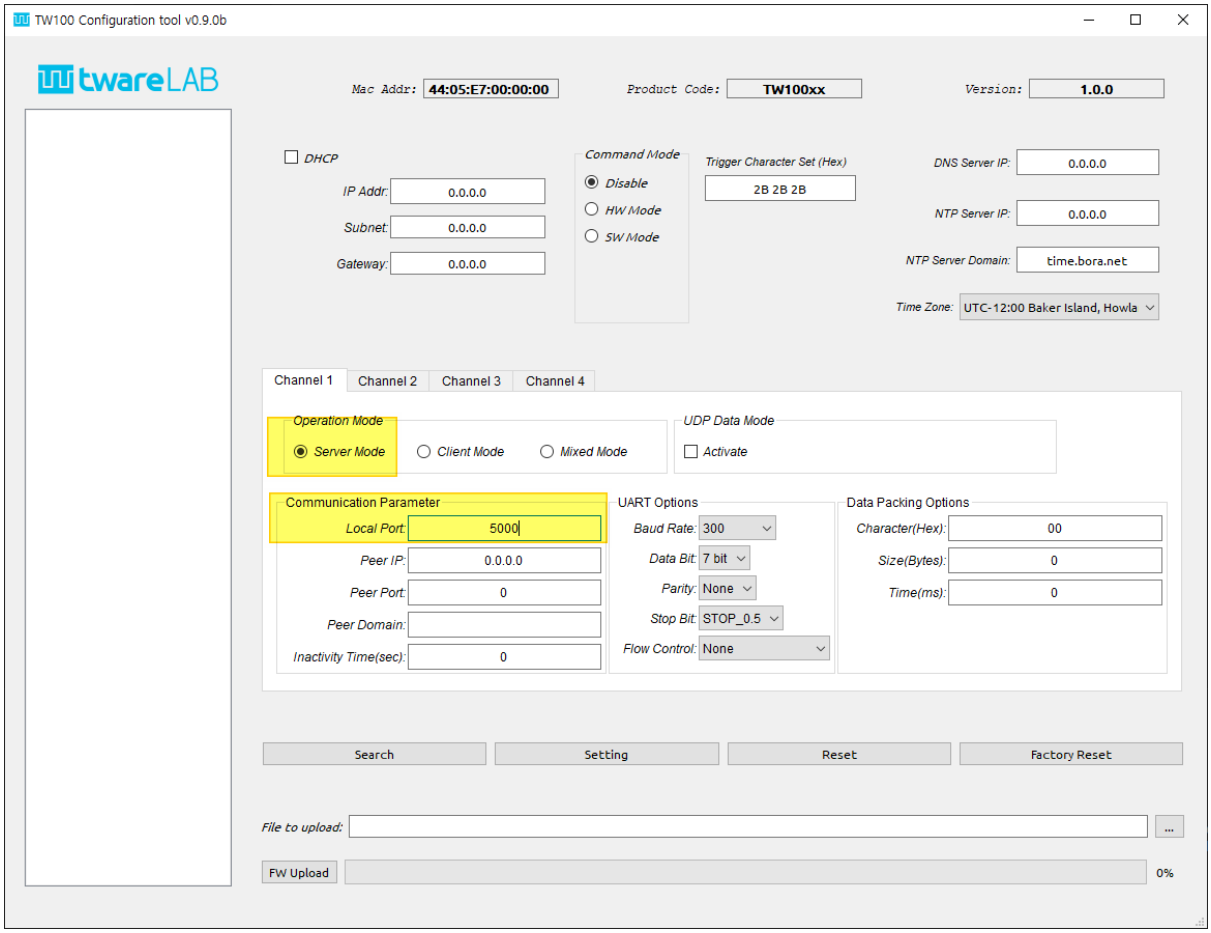

Figure 10 Server mode, Local Port settings

Usage example) After a user connects a TCP connection to TW100XX, let us take a look at the operation of sending data and sending it to the serial interface. (The Hercules program used in the example below supports both TCP/IP, UDP terminal functions, and serial port terminal functions, which is convenient for testing and can be downloaded from the following link. https://www.hw-group.com/software/hercules-setup-utility)

1. Attempt to connect to the TW100XX board (192.168.0.35:5000) and send "TEST_DATA_FROM_ETHERNET" data.

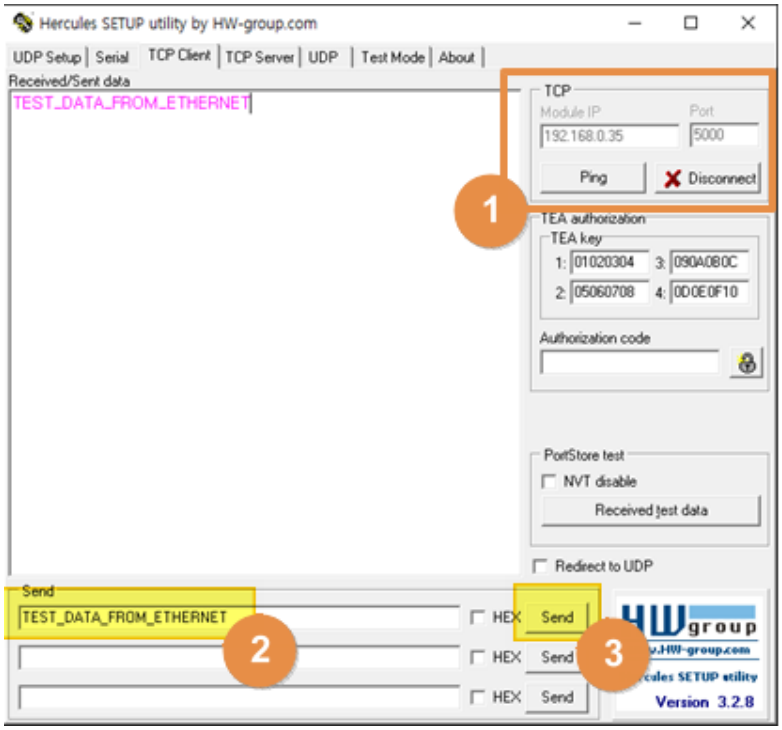

Figure 11 Data transfer after connecting to TCP client

2. You can check whether the "TEST_DATA_FROM_ETHERNET" data is received normally through the terminal of the serial interface (COM10, 115250bps) of the T W100XX board.

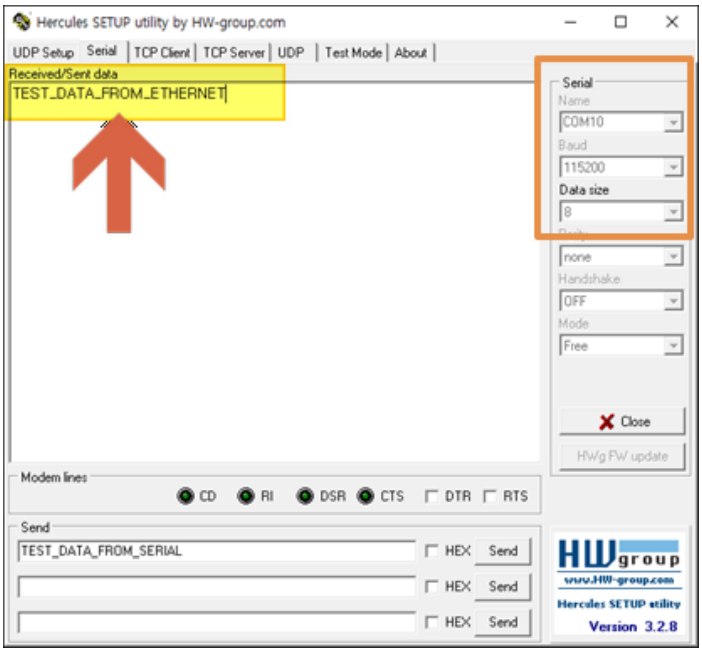

Figure 12 UART Terminal – Data Reception

6.2 TCP CLIENT MODE

Through the connected TCP connection, the method of transmitting and receiving data is the same. However, unlike the server mode, TX100XX module try to connect to the TCP server(Peer IP, Peer Port) until establishing the TCP connection.

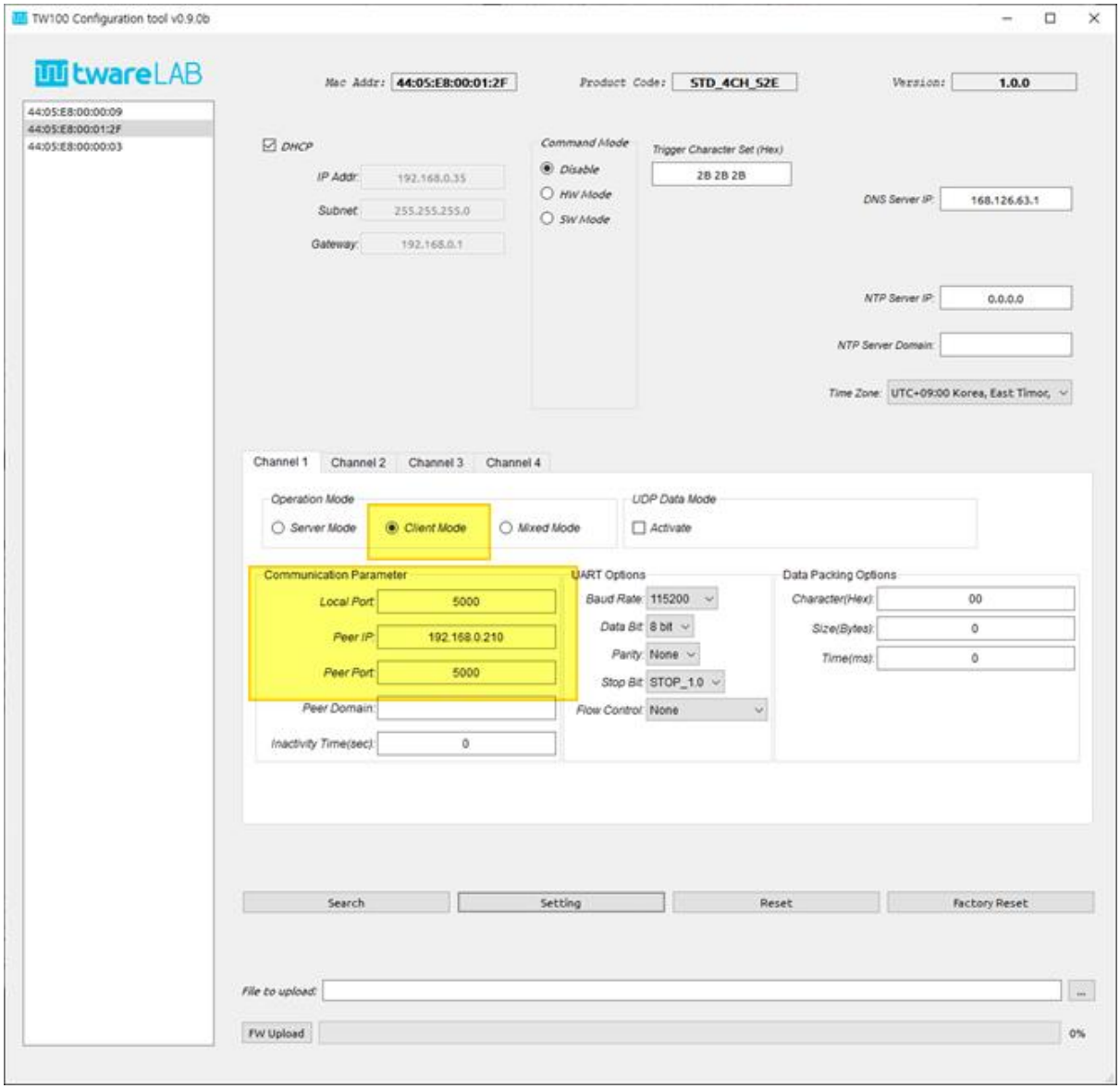

Figure 13 TCP Client mode, Peer IP/Port Settings

When the TCP connection is successful, it transmits the data received through the serial interface through the TCP connection, and sends the data received through the TCP connection to the serial interface.

6.3 TCP MIXED MODE

TCP Mixed mode is a method of Server and Client mode together. After the board reset, it operates in TCP Server mode and waits for a TCP connection attempt. In other words, it works the same as the TCP server mode. However, if serial data is received when there is no TCP connection, it is changed to Client mode without waiting as the server mode, and it supports trying to connect to TCP to the server specified in "Peer IP" and "Peer Port" and sending data immediately.

Therefore, when using Mixed mode, all the values of "Operation Mode", "Local Port", "Peer IP", and "Peer Port" must be set as shown in the figure below.

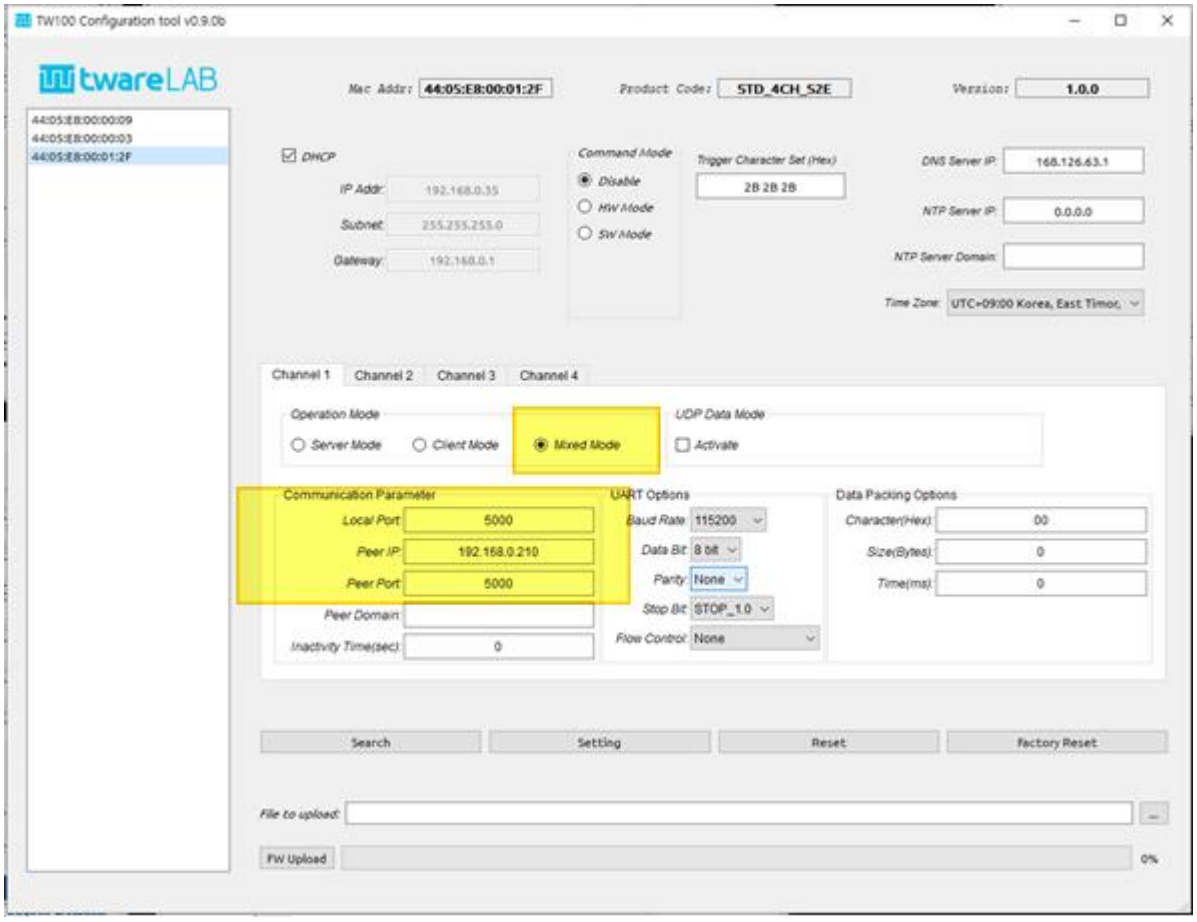

Figure 14 TCP Mixed mode, Local Port, Peer IP/Port Settings

6.4 UDP Mode

UDP mode is a mode that sends serial data through UDP communication instead of TCP.

The configuration is similar to TCP Mixed mode, as shown in the figure below. However, UDP Data Mode must be "Activate".

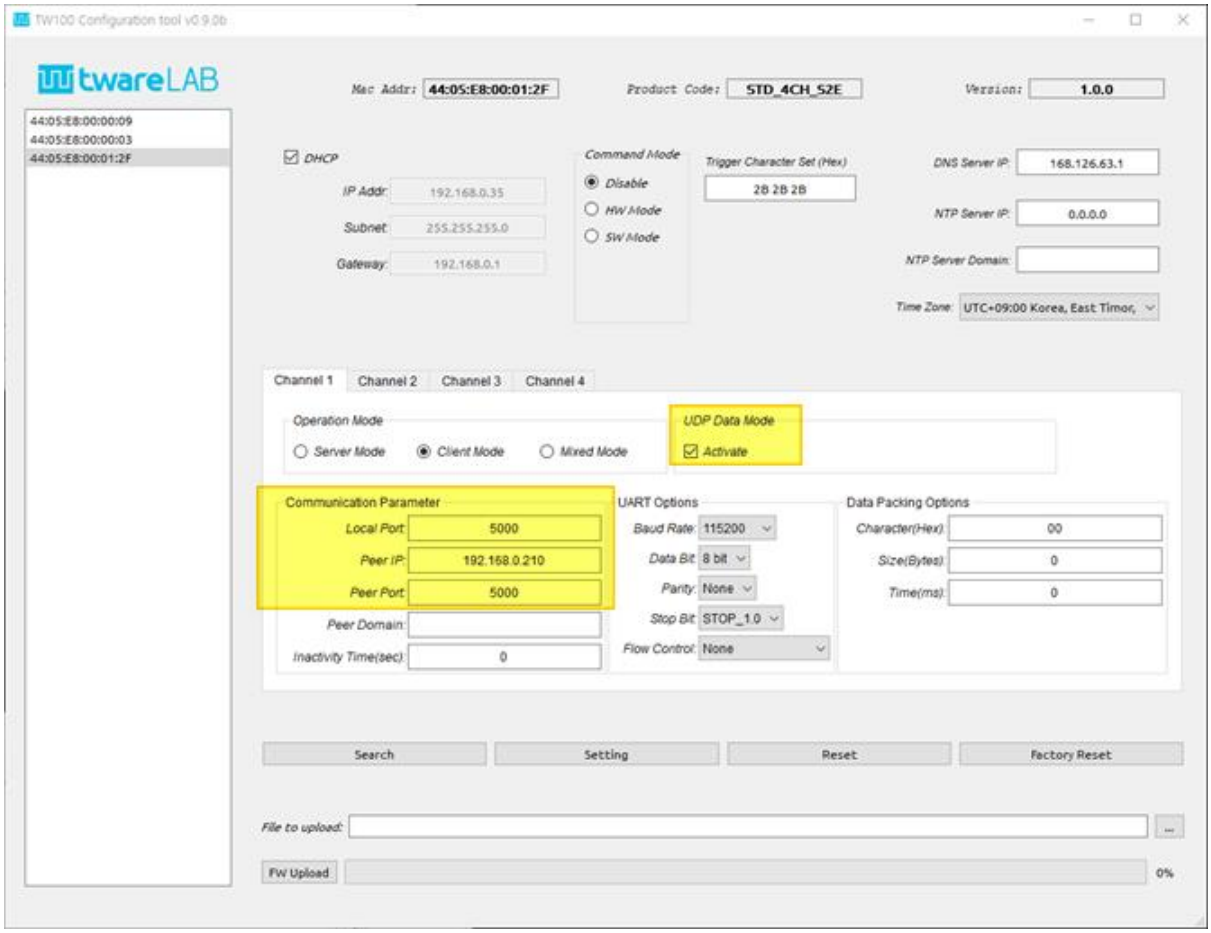

Figure 15 UDP mode, Local Port, Peer IP/Port settings

Usage example) When the user enables UDP mode on TW100XX and sends serial data, it is sent to UDP.

1. Sends data to the serial interface connected to the TW100XX.

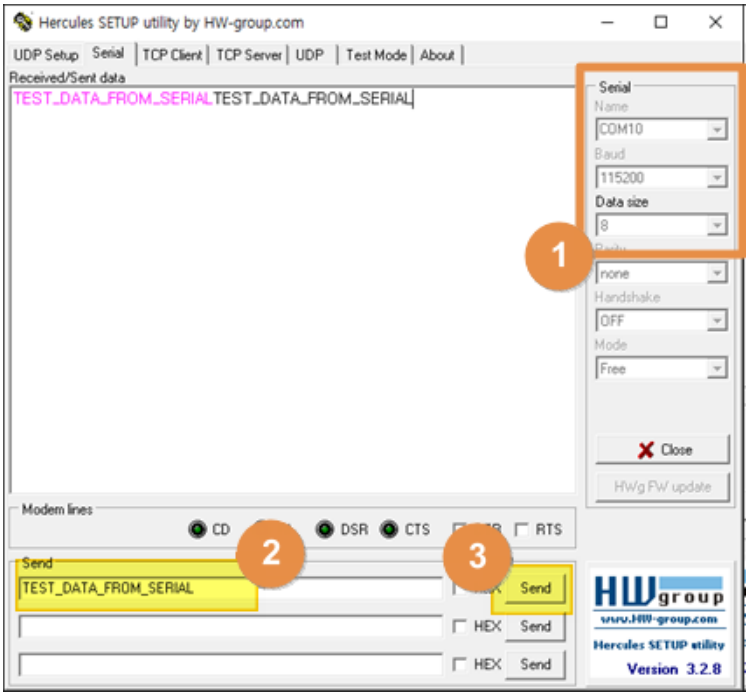

Figure 16 UART connection and data transfer

2. As shown in the figure below, you can see that data is normally received via UDP through the network terminal.

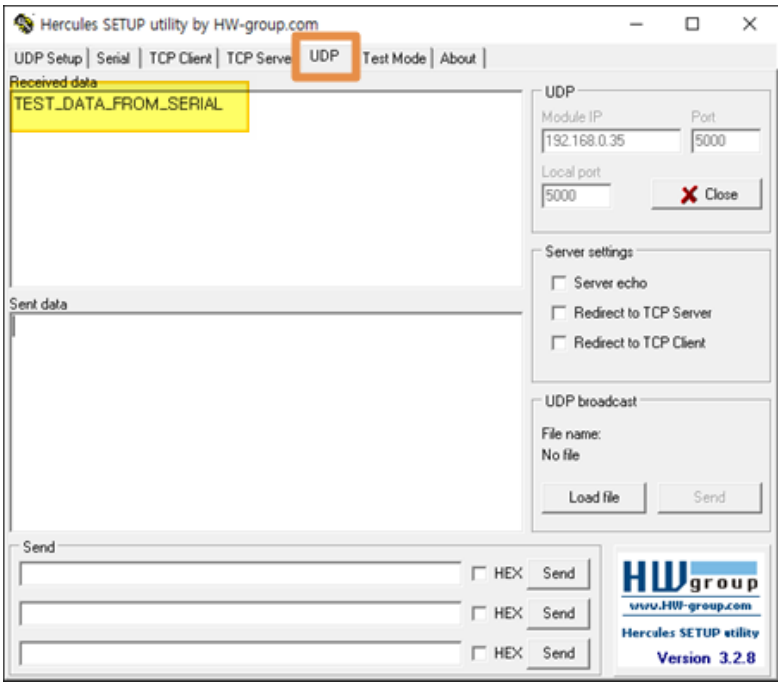

Figure 17 UDP Data Reception

6.5 DATA PACKING OPTION

When transmitting serial data to the TCP/IP network, the basic method is to send the data received from the serial directly in real time, but it is a function used when you want to collect data until a specific condition is satisfied and send the data to the TCP/IP network all at once. If two or more conditions are set, data is transmitted as soon as one condition is satisfied.

① Character (Hex code) – Collects serial buffer data until a specified character is entered, and then creates and transmits Ethernet packets at a time. In the Configuration tool, you can specify it as a hex code value. For example, if you want to set a carriage return(CR), enter "0d", which is the hex code value of ascii code. If the value is "00", disable this condition.

② Size (0 ~ 1000 bytes) - Wait until a certain length of data is stored in the serial buffer, and when the data bytes of that length are collected, they are made into Ethernet packets and transmitted at once. If the value is "0", disables this condition.

③ Time (0 ~ 65535 ms) – Collects serial buffer data until the specified time (milliseconds) is reached, and then creates and transmits Ethernet packets at a time. If the value is "0", disables this condition.

Usage example) Let us look at the behavior of TW100XX when the user activates the "Character" part of the "Data Packing Condition" section.

① Set the value of 0x0A in the “Character" section of the "Data Packing Option" section.

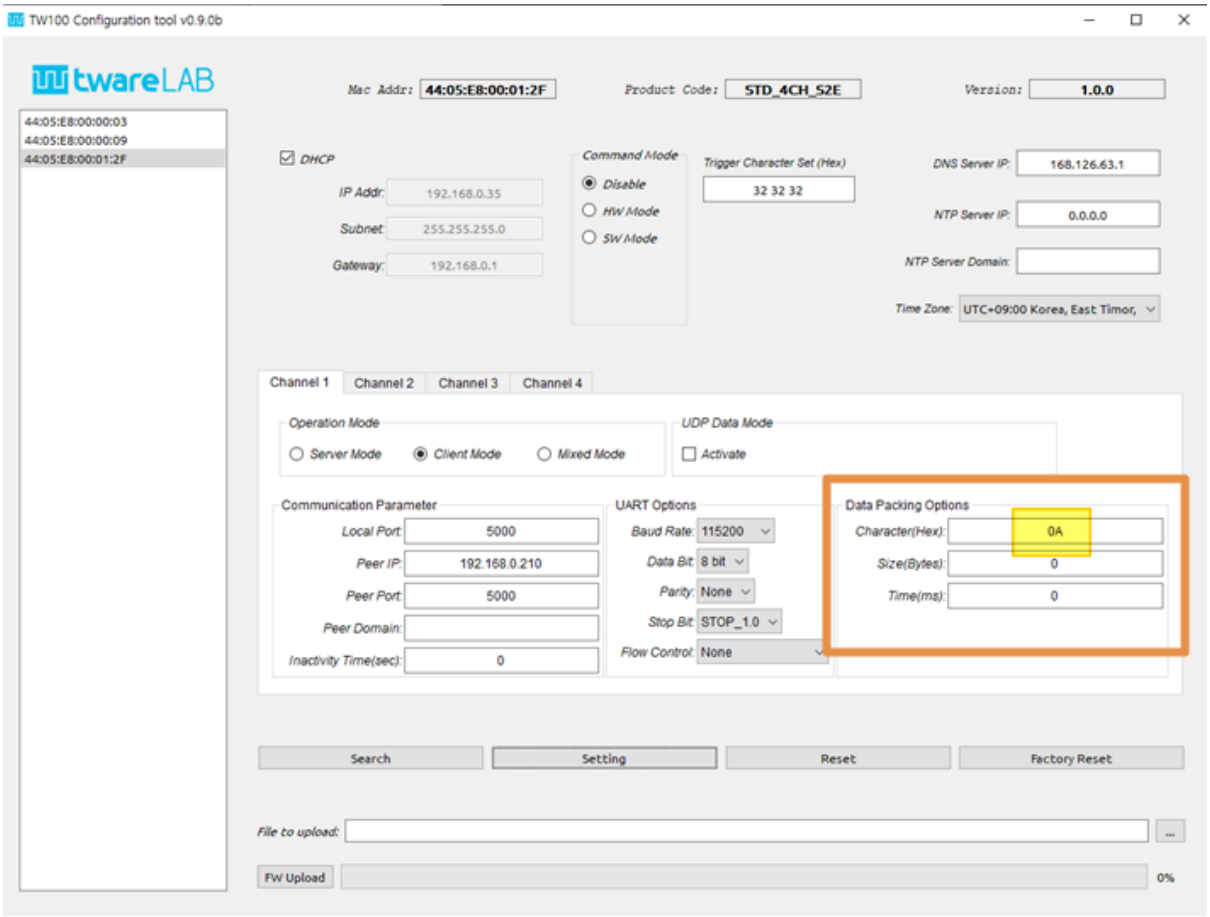

Figure 18 Data packing – Character setting

② Send data containing 0x0A to the serial interface connected to the T W100XX as shown below. *"1234567" + 0x0A*

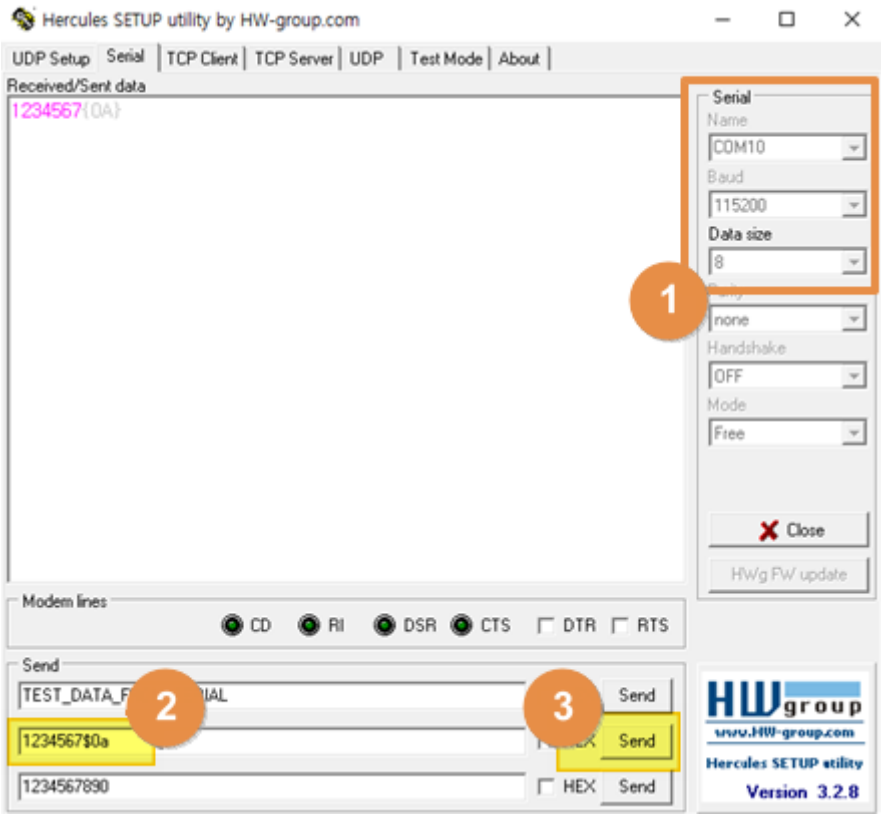

Figure 19 UART Data Transfer (including Data Packing character)

③ In PC connected to TCP, the "*1234567*" *+0x0A* value is normally entered at once. (If the Data Packing Option is not set, it is received in several data packets instead of at once.)

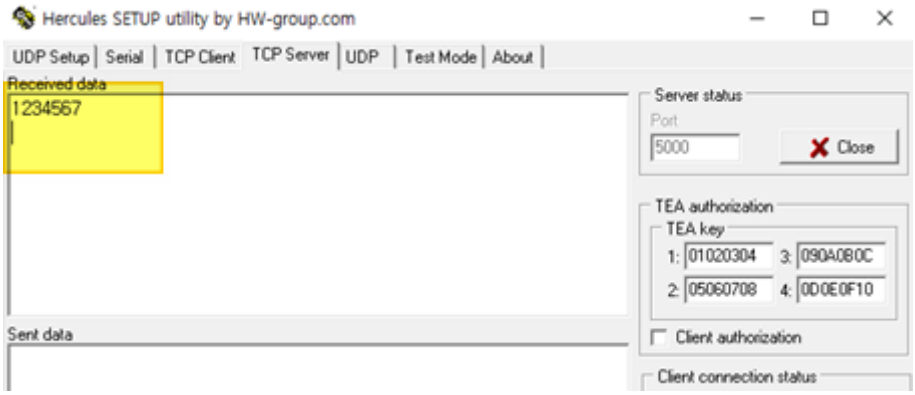

Figure 20 Receiving Packing Data

6.5.1 Data Packing Option Order

The Data Packing Option can be used in combination of three conditions. The order of conditions is Character -> Size -> Time.

7 EVALUATION BOARD

7.1 HARDWARE SPECIFICATION

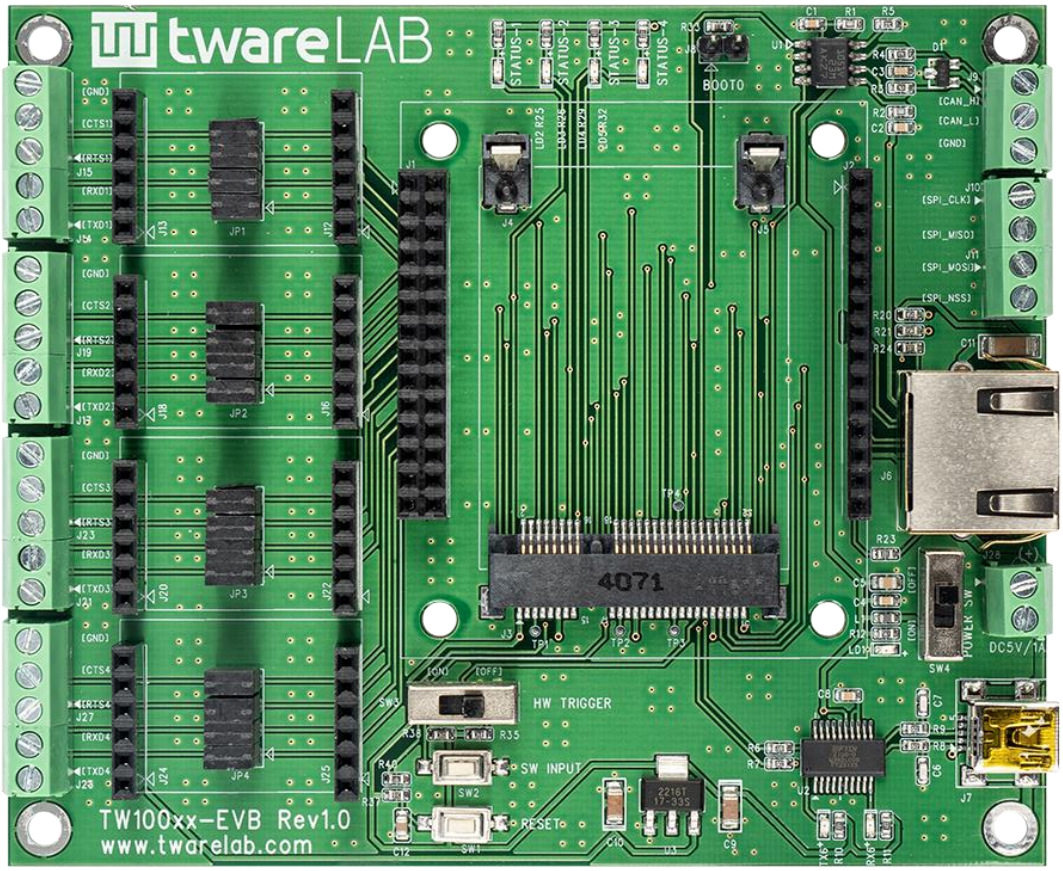

Figure 21 TW100xx EVB Image

The TW100xx-EVB is divided into 6 blocks.

Serial terminal blocks, CAN/SPI terminal blocks, power supply blocks, external switch input blocks, module mounting blocks, and Ethernet connection LED blocks. Each block is described in the following sections.

7.1.1 Serial Terminal Block

The serial terminal block is a terminal block for connecting with each serial port for data communication from the outside. Basically, the serial terminal is connected to the UART port of the TW100xx module at a 3.3V TTL level.

Four RXD/TXD/RTS/CTS pins are connected.

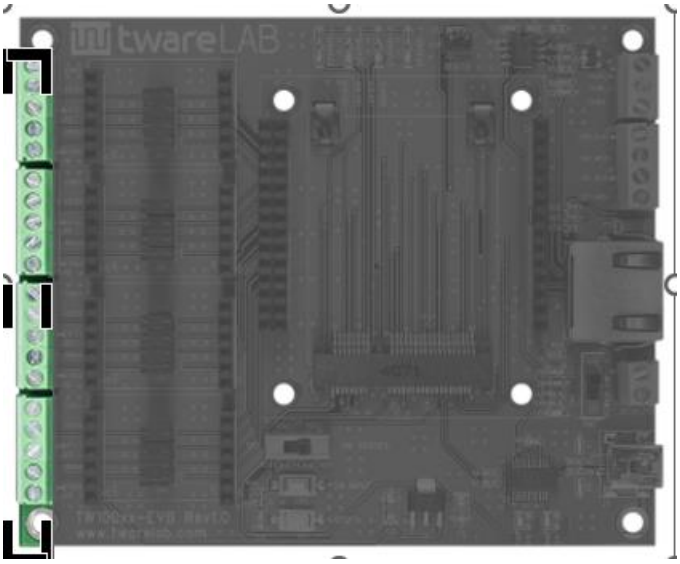

Figure 22 Serial Terminal Block

7.1.1.1 Signal Level Converter

Serial Converter Modules can be mounted on the header sockets inside each terminal to switch to RS232/RS485/RS422 signal levels.

To convert signal levels by installing a serial converter module, remove the shunter between each header socket and mount the appropriate Protocol converter module into the header socket.

The table below shows which signal each pin represents depending on whether the Serial Converter Modules module is used or not.

Silk markings	3.3V TTL	RS232	RS485	RS422
CTS	CTS	CTS(232 level)	*NC*	TX+

RTS	RTS	RTS(232 level)	*NC*	TX-
RXD	RXD	RXD(232 level)	485+	RX+
TXD	TXD	TXD(232 level)	485-	RX-

Table 14serial terminal signals when using the Protocol converter

7.1.2 CAN & SPI Terminal Block

CAN and SPI terminals will be supported in the future for connecting for CAN communication or SPI communication.

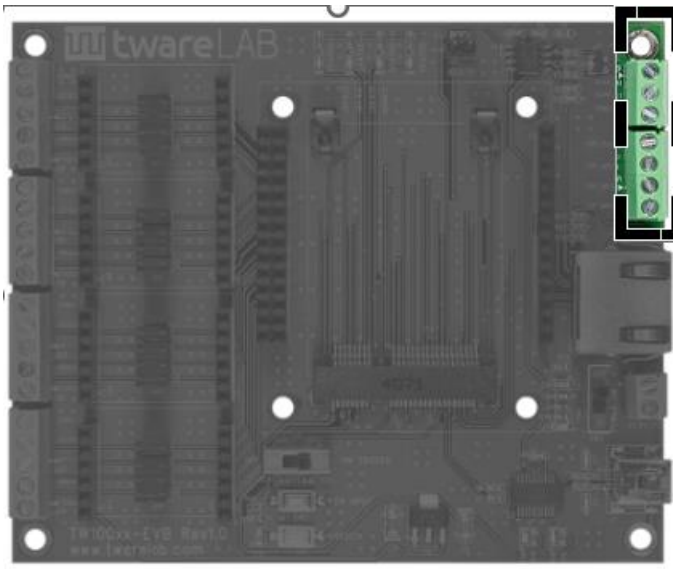

Figure 23 CAN Terminal and SPI Terminal

7.1.3 Power Input Block

The power input block includes a power ON/OFF switch for connecting external power supply. The external power input can be directly connected to a 5V power supply or 5V can be supplied from a PC via a USB socket.

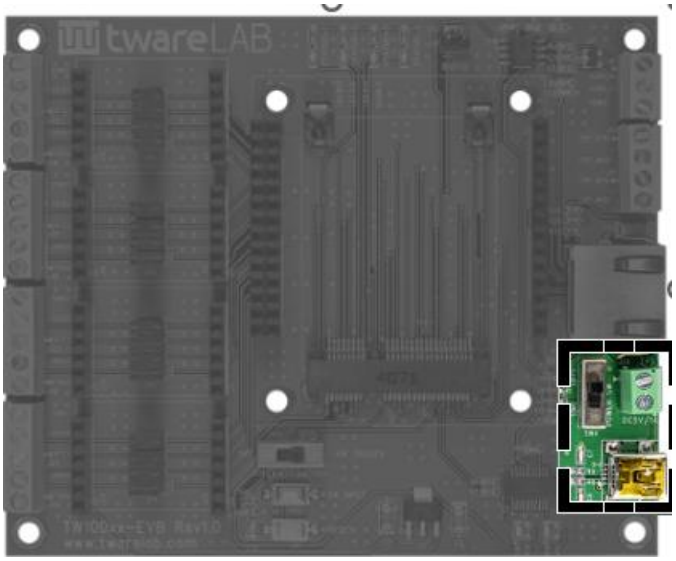

Figure 24 Power Input Block

7.1.4 Reset & Switch Block

This block consists of a Push Switch (SW1) for resetting the Hardware module, a Push Switch (SW2) and a Slide Switch (SW3) for input external options.

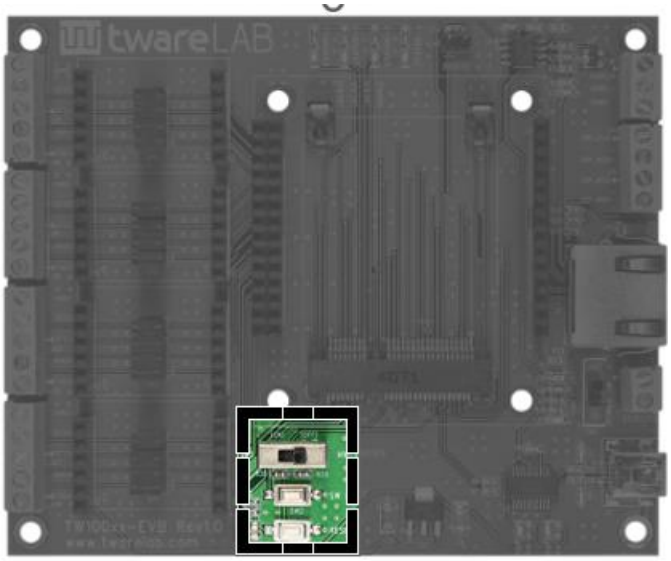

Figure 25 External Input Block

The functions of each Switch are as follows.

category	use	Details
SW1	Module Hardware Reset	
SW2	SW INPUT	Software Reset/Factory Reset (**Reset**)
SW3	HW Trigger	AT Command Mode Enable/Disable

Table 15 External Input Switch Functions

7.1.5 S2E Module Socket Block

This block is for mounting either TW100MJ/XR/PC.

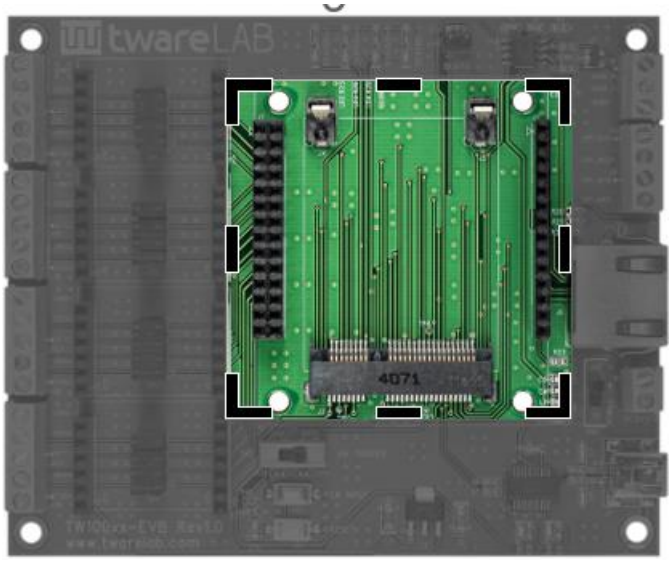

Figure 26 TW100xx Socket Block

7.1.6 Status LED Block

This block indicates a TCP connection status of each socket through an LED. When a TCP connection is established, the corresponding LED is ON, and when the connection is closed, the LED is OFF.

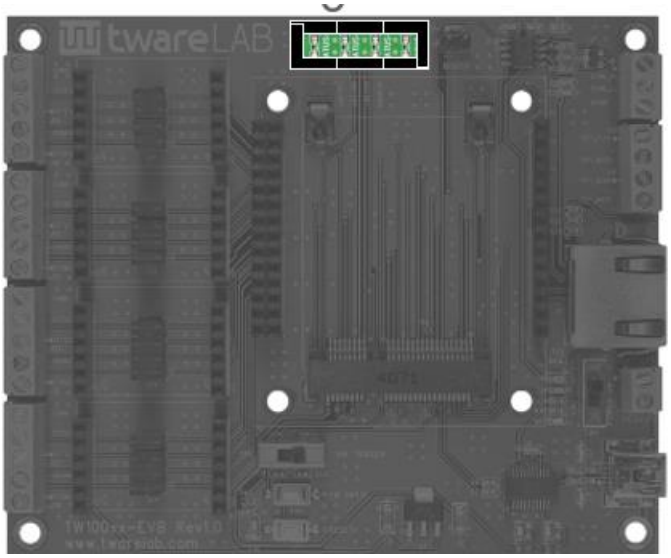

Figure 27 Socket Connection Indicating LEDs

7.2 TW100MJ EXAMPLE

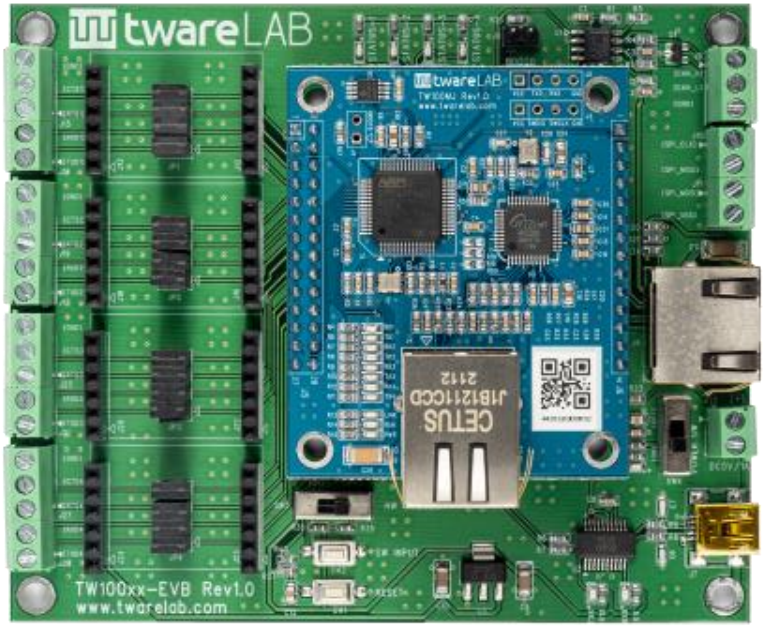

Figure 28 TW100MJ Plugged EVB

7.3 TW100XR EXAMPLE

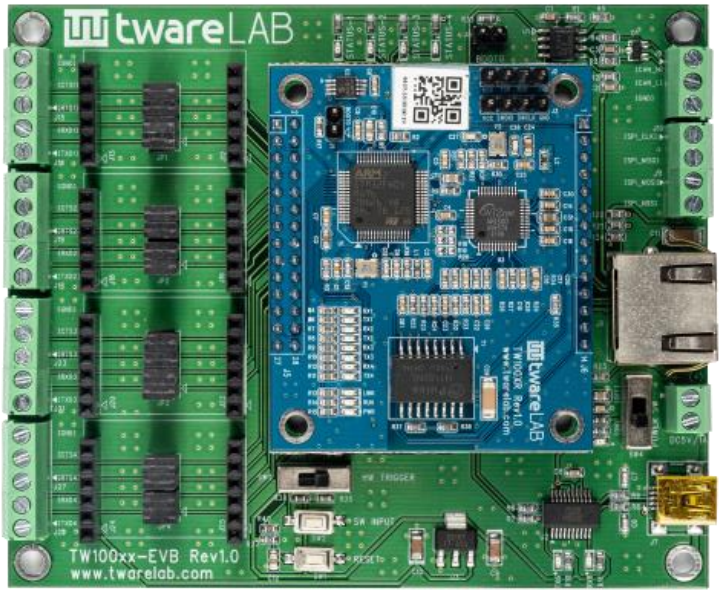

Figure 29 TW100XR Plugged EVB

7.4 TW100PC EXAMPLE

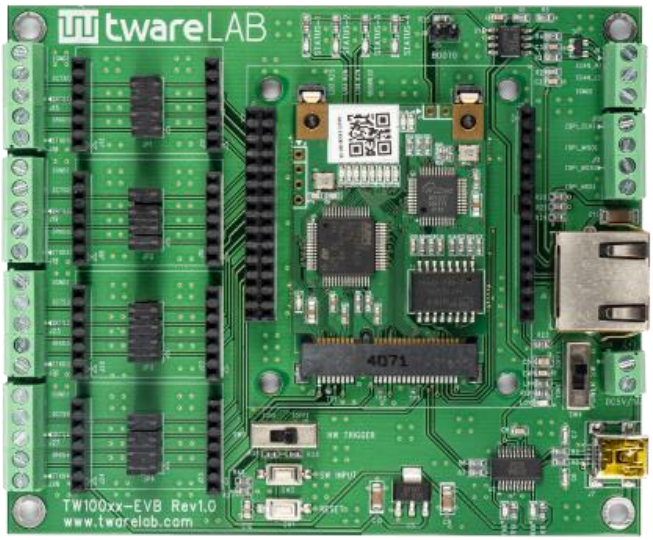

Figure 30 TW100PC Plugged EVB

7.5 Serial Converter Modules

7.5.1 TTL-to-232 Module

Figure 31 TTL-to-232 Converter Module

7.5.2 TTL-to-485 Module

Figure 32 TTL-to-RS485 Converter Module

7.5.3 TTL-to-422 Module

Figure 33 TTL-to-RS422 Converter Module

7.5.4 Converter Module connection

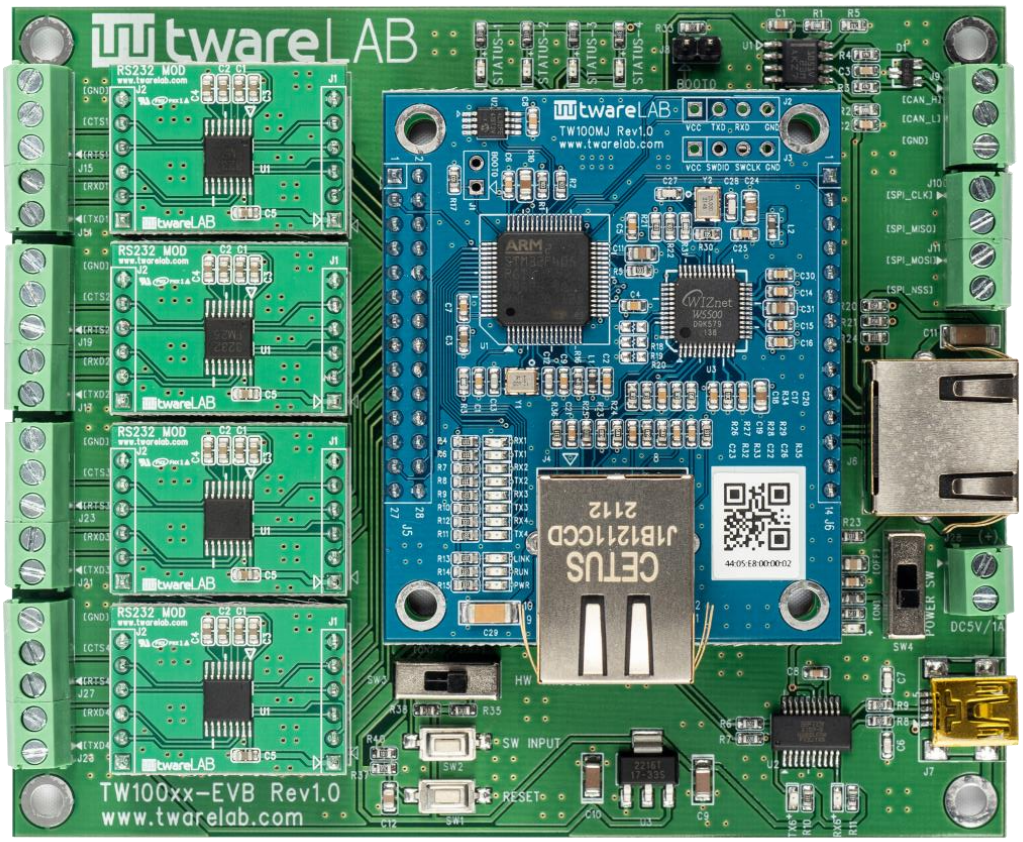

Figure 34 TW100MJ and TTL-to-RS232 modules plugged EVB

8 HISTORY

Date	Description
2022-11-02	V1.0 First Released
2022-11-09	PeerInfo Command Error Correction