AN0052
Application Note

AT32 EMAC Client and Server

Introduction

As required by the development of IoT, devices including chips are designed with the ability to connect to the Internet. In response to this demand, AT32 microcontrollers support EMAC feature.

This application note, taking AT32F407 series as an example, gives examples to help users develop features as needed.

Applicable products:

Part number	AT32 MCUs with EMAC feature

Contents

List of tables

List of figures

1 Overview

TCP Server and TCP Client are quite different during communication. A TCP Server waits for TCP connections from TCP clients through a specific port, and a TCP client issues a connection request to the TCP server. In this routine, LwIP protocol stack is used, and three system calls, i.e., tcp_bind(), tcp_listen() and tcp_accept(), are used by TCP Server to accept the connection request from Client.

This application note introduces how TCP client and TCP server communicate with PC.

1.1 Hardware requirements

1) AT-START-F407 V1.0 evaluation board
2) DM9162 Ethernet module
3) Ethernet cable

1.2 Software requirements

- tcp_client, TCP client source program, run TCP client programs
- tcp_server, TCP server source program, run TCP server programs

2 AT32 TCP client/server program settings

2.1 Pin settings

Table 1. Pin configuration

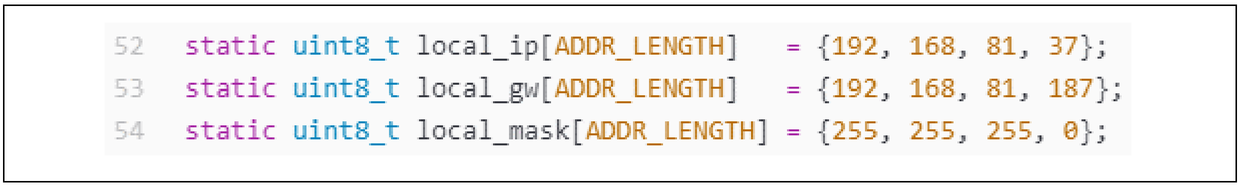
EMAC signal	Pin
EMAC_MDC	PC1
EMAC_MDIO	PA2
EMAC_RMII_REF_CLK	PA1
EMAC_RMII_CRS_DV	PD8
EMAC_RMII_RXD0	PD9
EMAC_RMII_RXD1	PD10
EMAC_RMII_TX_EN	PB11
EMAC_RMII_TXD0	PB12
EMAC_RMII_TXD1	PB13

2.2 LwIP settings

Hardware handles signals from PHY to MAC only. For further development, users need to implement TCP/IP protocol stack. In this example, LwIP protocol stack is used to reduce memory usage and program code size so that the LwIP can be used for resource-limited platforms (such as embedded systems). For more details, please visit the official website.

The protocol stack is completely integrated into the code; therefore, users only need to set the IP address and date address according to the network segments. These two global variables are declared in netconf.c.

Figure 1. Configure MCU IP address

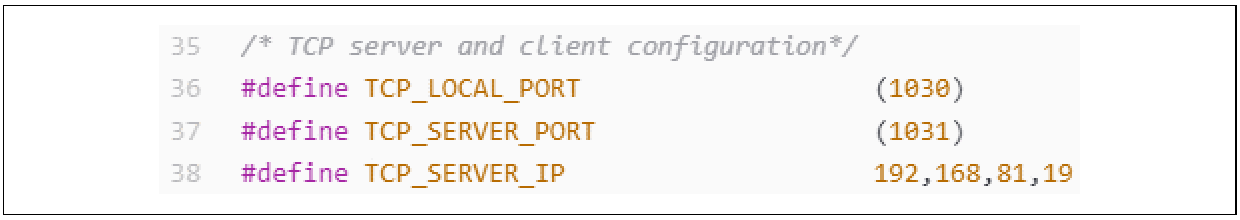


```
static uint8_t local_ip[ADDR_LENGTH]   = {192, 168, 81, 37};
static uint8_t local_gw[ADDR_LENGTH]   = {192, 168, 81, 187};
static uint8_t local_mask[ADDR_LENGTH] = {255, 255, 255, 0};
```

2.3 TCP client project settings

Initialize LwIP protocol stack and then configure TCP client. The chip serves as the Client to issue a request, and users need to configure the server's socket that is used for handling incoming connections. The socket is the IP address plus a port. There are several macro definitions declared in the code to represent server's sockets. For example, TCP_SERVER_IP and TCP_SERVER_PORT macro definitions are contained in tcp_client.h, and they can be modified according to the segment and application requirements.

Figure 2. Configure TCP server socket



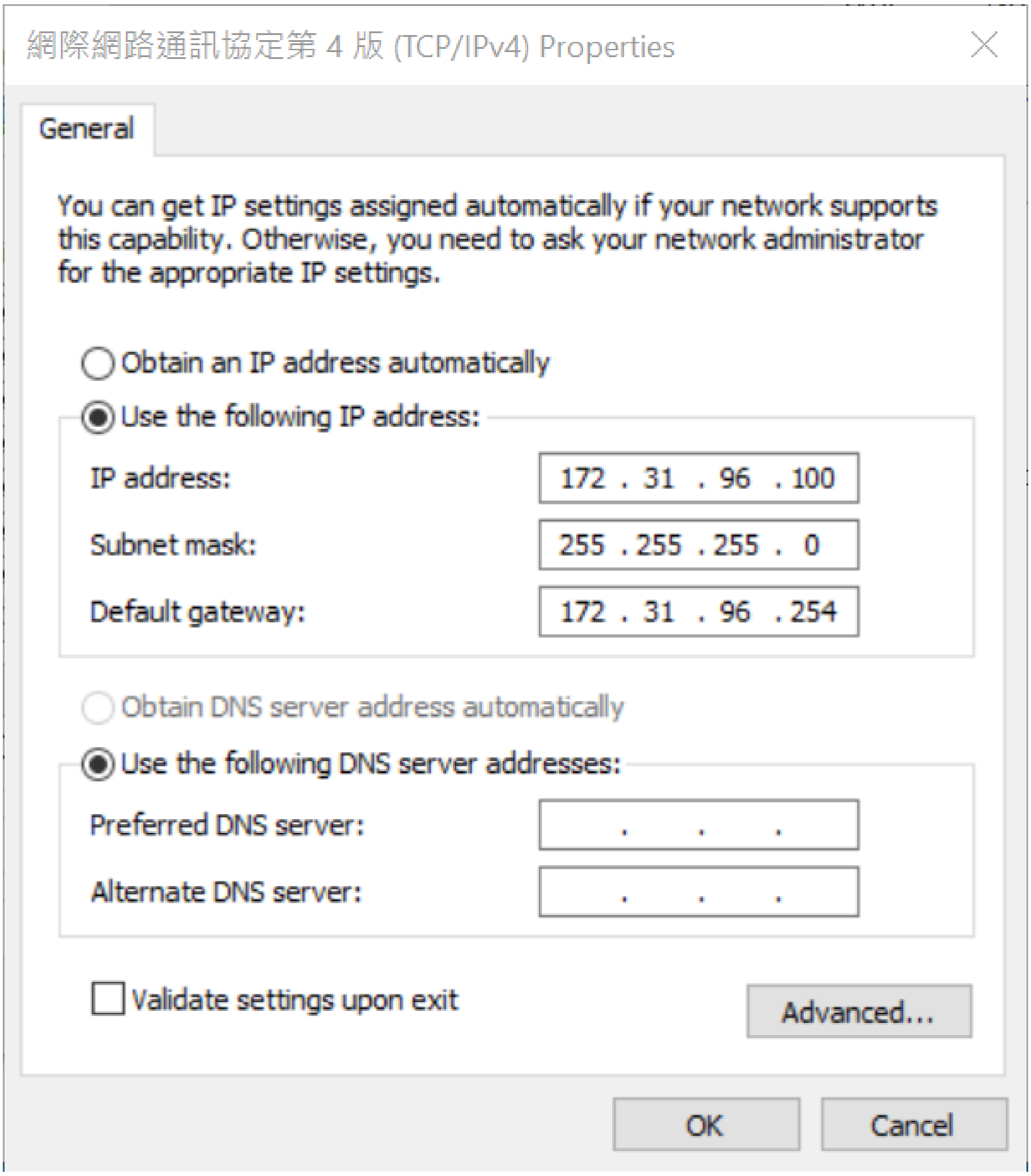
```
/* TCP server and client configuration*/
#define TCP_LOCAL_PORT                    (1030)
#define TCP_SERVER_PORT                   (1031)
#define TCP_SERVER_IP                     192,168,81,19
```

Ensure that the network segment is the same as that on the server, and then download code to the chip. The server will print “tcp client experiment!” once a second, and LED2, LED3 and LED4 will blink.

2.4 Host TCP Server settings

1) Set the host IP address, network mask and gate. The IP address and gate should be in the same network segment as the chip.
2) Open the host server software (here use network debugging assistant), select TCP server, local IP address is TCP_SERVER_IP and local port number is TCP_SERVER_PORT. Then, click Connect to establish connection.
3) Once a connection is established, the below interface pops up, printing data from the TCP client.

Figure 3. Set PC IP, network mask and gate



網際網路通訊協定第 4 版 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 31 . 96 . 100

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 172 . 31 . 96 . 254

○ Obtain DNS server address automatically

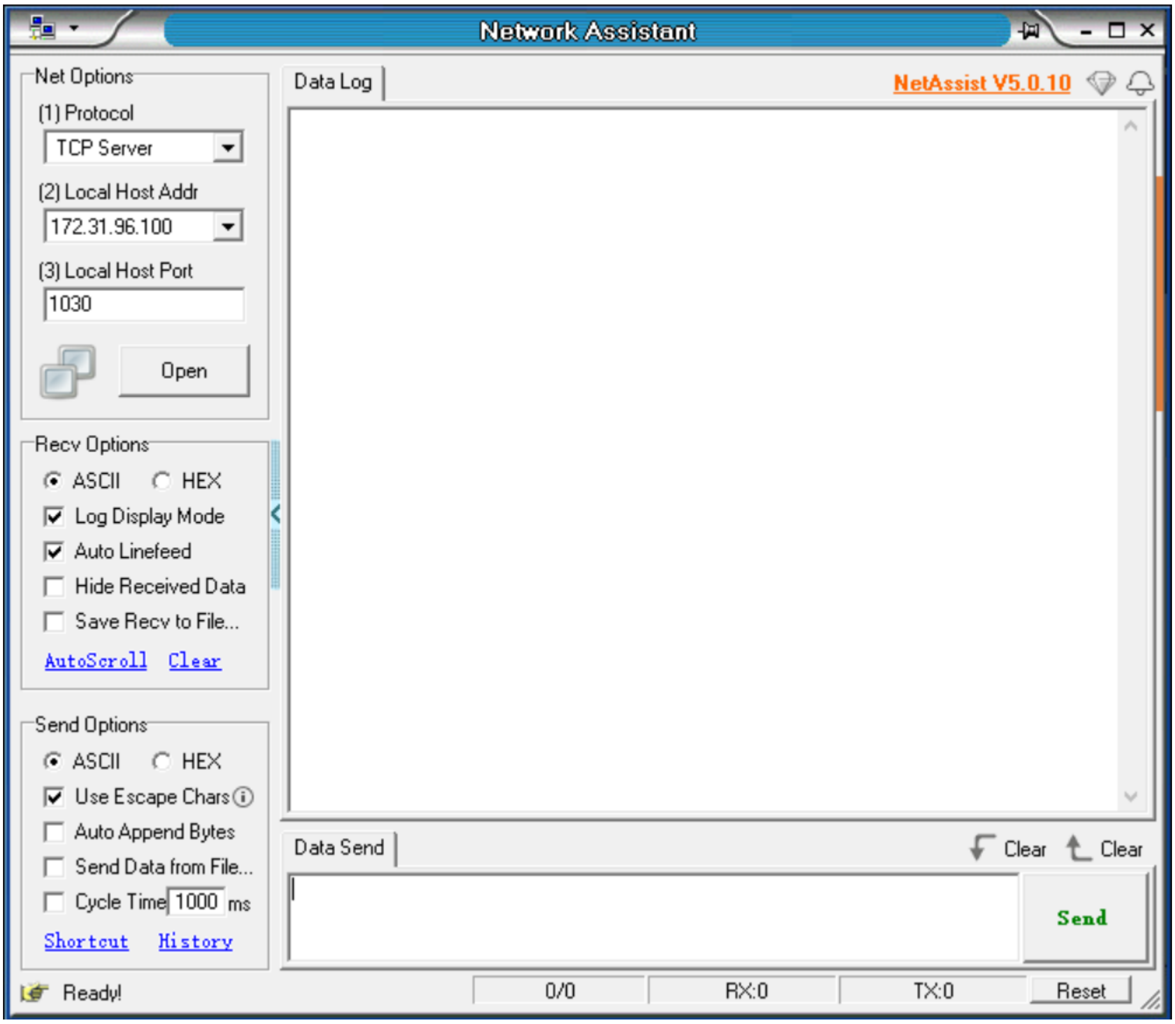
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Figure 4. Set server software



Net Options

(1) Protocol: TCP Server

(2) Local Host Addr: 172.31.96.100

(3) Local Host Port: 1030

Recv Options

● ASCII ○ HEX

☑ Log Display Mode

☑ Auto Linefeed

□ Hide Received Data

□ Save Recv to File...

Send Options

● ASCII ○ HEX

☑ Use Escape Chars

□ Auto Append Bytes

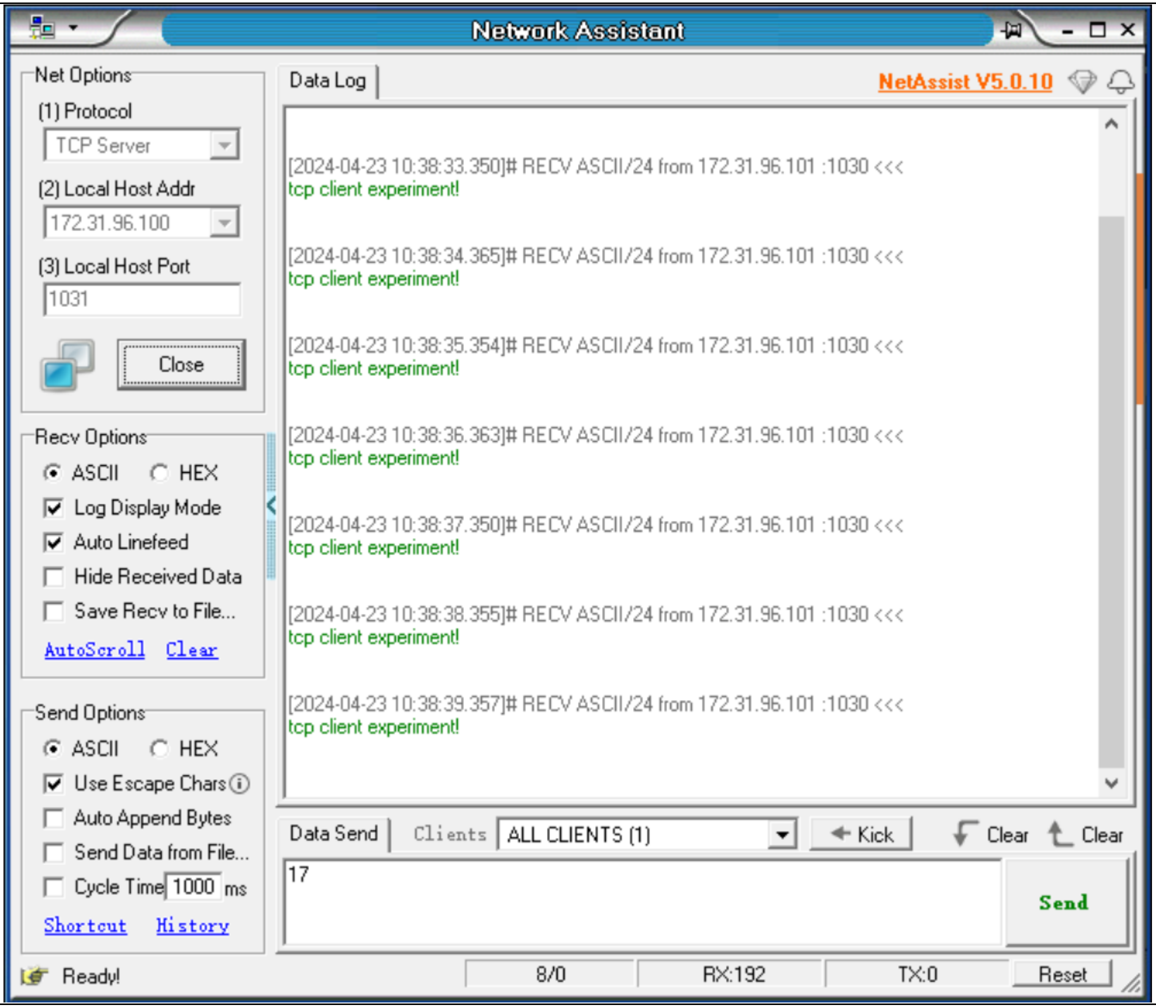
□ Send Data from File...

□ Cycle Time 1000 ms

Ready!

0/0 RX:0 TX:0

Figure 5. Receive data from TCP client



Network Assistant

Net Options

(1) Protocol: TCP Server

(2) Local Host Addr: 172.31.96.100

(3) Local Host Port: 1031

Recv Options

- ● ASCII ○ HEX
- ☑ Log Display Mode
- ☑ Auto Linefeed
- □ Hide Received Data
- □ Save Recv to File...

Send Options

- ● ASCII ○ HEX
- ☑ Use Escape Chars
- □ Auto Append Bytes
- □ Send Data from File...
- □ Cycle Time 1000 ms

Data Log — NetAssist V5.0.10

```
[2024-04-23 10:38:33.350]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:34.365]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:35.354]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:36.363]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:37.350]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:38.355]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!

[2024-04-23 10:38:39.357]# RECV ASCII/24 from 172.31.96.101 :1030 <<<
tcp client experiment!
```

Data Send — Clients: ALL CLIENTS (1)

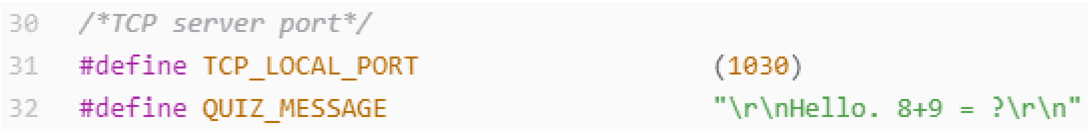
17

Ready! 8/0 RX:192 TX:0

2.5 TCP server project settings

Initialize LwIP protocol stack and then configure TCP server. The TCP server receives connection requests from the client, which requires a port for the client to send data. A micro definition is declared in the code to represent the server, that is, the TCP_LOCAL_PORT in tcp_server.h. This macro definition can be modified as needed.

Figure 6. TCP server port macro definition

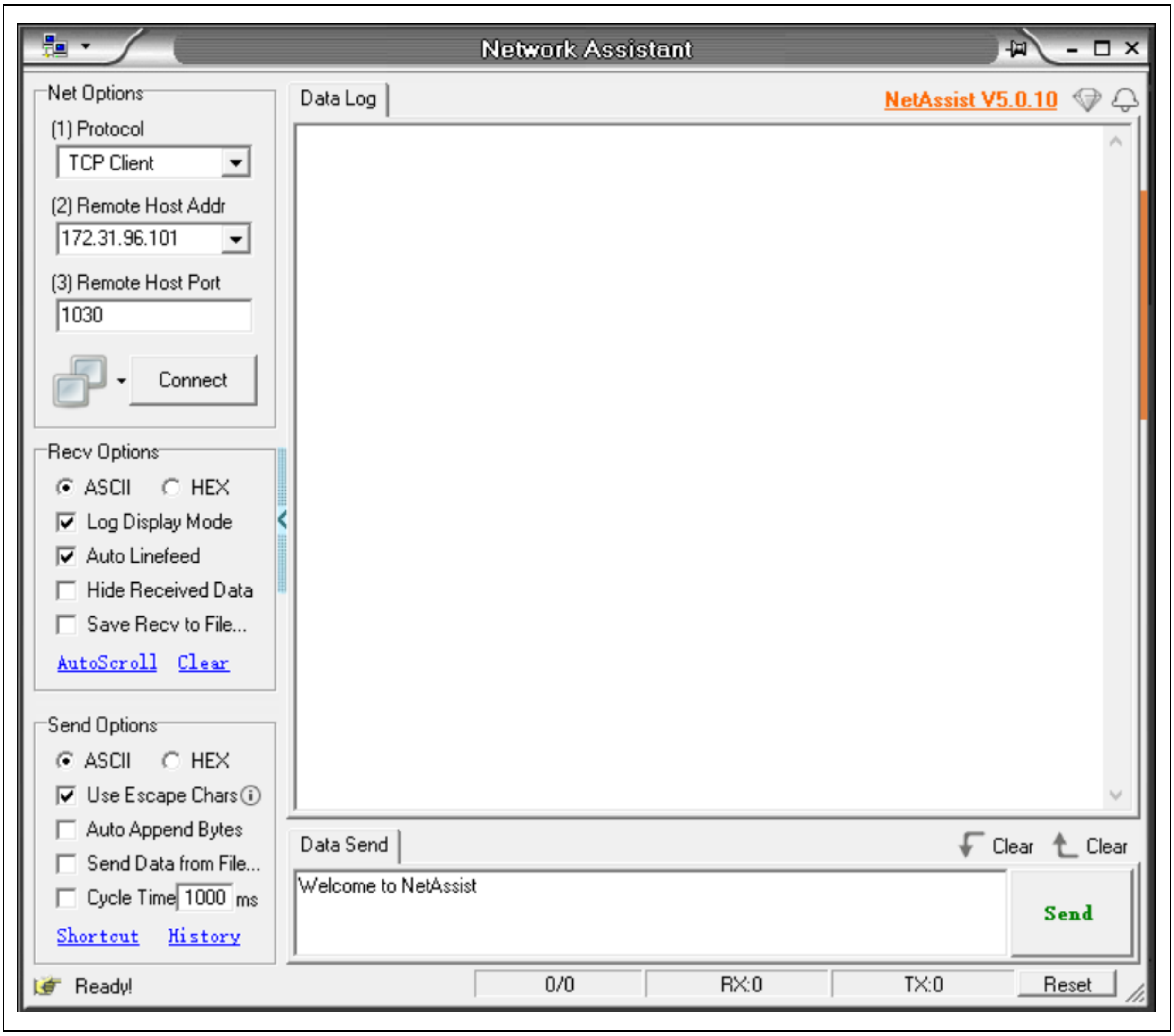


```
/*TCP server port*/
#define TCP_LOCAL_PORT                   (1030)
#define QUIZ_MESSAGE                     "\r\nHello. 8+9 = ?\r\n"
```

2.6 Host TCP Client settings

1) Set the host IP address, network mask and gate. The IP address and gate should be in the same network segment as the chip, as shown in Figure 3.
2) Open the host client software (here use network debugging assistant), select TCP client, server IP address is the chip IP address, and local port number is TCP_LOCAL_PORT. Then, click Connect to establish connection.
3) At this point, input strings into the bottom block and send to the server, and then the server will respond to the input strings, as shown in Figure 8.

Figure 7. Set host client software



Network Assistant

Net Options

(1) Protocol: TCP Client

(2) Remote Host Addr: 172.31.96.101

(3) Remote Host Port: 1030

Recv Options

● ASCII ○ HEX

☑ Log Display Mode

☑ Auto Linefeed

□ Hide Received Data

□ Save Recv to File...

Send Options

● ASCII ○ HEX

☑ Use Escape Chars

□ Auto Append Bytes

□ Send Data from File...

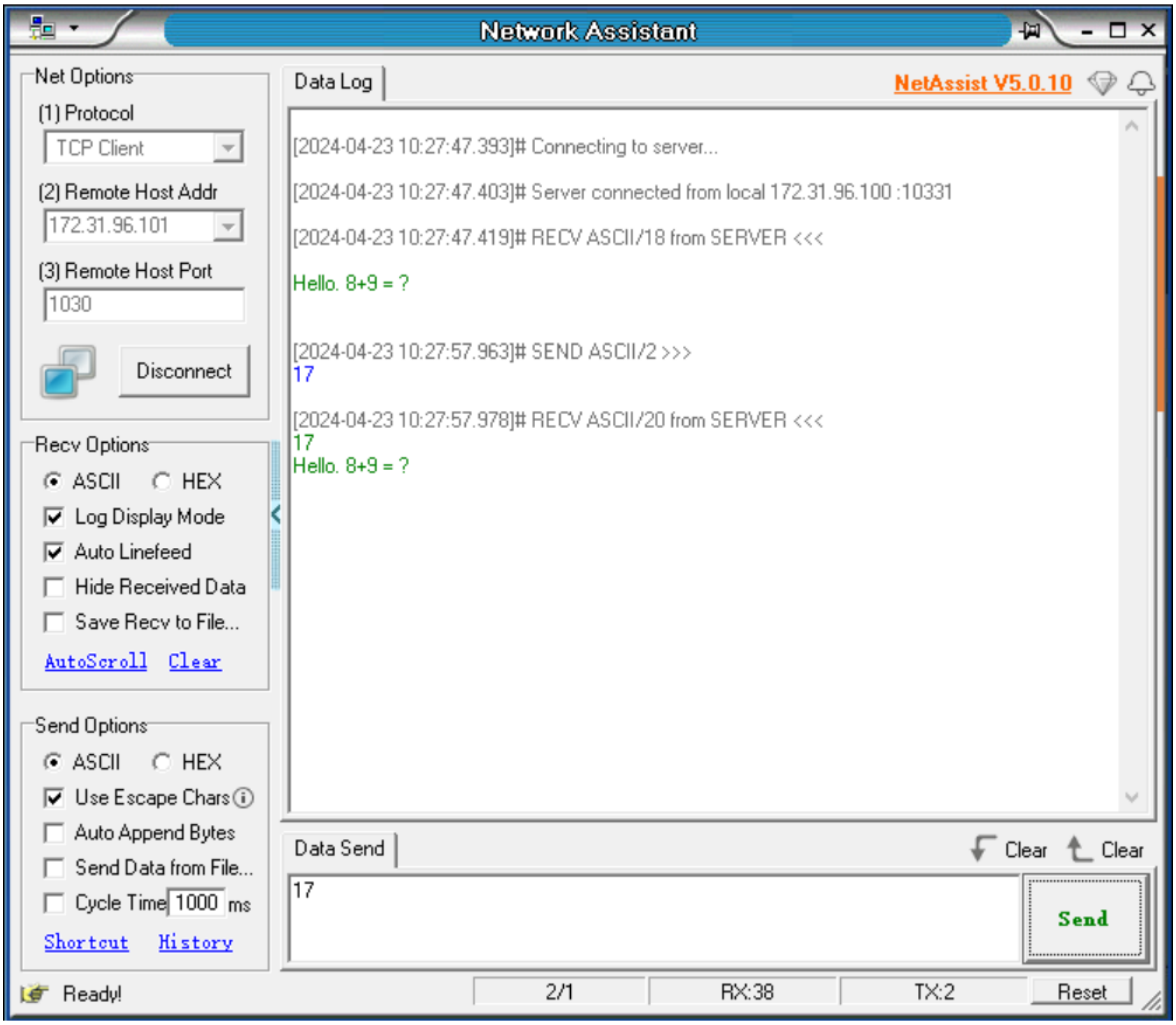
□ Cycle Time 1000 ms

Data Send: Welcome to NetAssist

Ready!

0/0 RX:0 TX:0

Figure 8. Data display window and response to input strings



Net Options
(1) Protocol: TCP Client
(2) Remote Host Addr: 172.31.96.101
(3) Remote Host Port: 1030

Recv Options
● ASCII ○ HEX
☑ Log Display Mode
☑ Auto Linefeed
□ Hide Received Data
□ Save Recv to File...

Send Options
● ASCII ○ HEX
☑ Use Escape Chars
□ Auto Append Bytes
□ Send Data from File...
□ Cycle Time 1000 ms

```
[2024-04-23 10:27:47.393]# Connecting to server...
[2024-04-23 10:27:47.403]# Server connected from local 172.31.96.100 :10331
[2024-04-23 10:27:47.419]# RECV ASCII/18 from SERVER <<<
Hello. 8+9 = ?

[2024-04-23 10:27:57.963]# SEND ASCII/2 >>>
17

[2024-04-23 10:27:57.978]# RECV ASCII/20 from SERVER <<<
17
Hello. 8+9 = ?
```

Data Send: 17

2/1 RX:38 TX:2

3 Revision history

Table 2. Document revision history

Date	Version	Revision note
2021.09.03	2.0.0	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services; ARTERY assumes no liability for purchasers' selection or use of the products and the relevant services.

No license, express or implied, to any intellectual property right is granted by ARTERY herein regardless of the existence of any previous representation in any forms. If any part of this document involves third party's products or services, it does NOT imply that ARTERY authorizes the use of the third party's products or services, or permits any of the intellectual property, or guarantees any uses of the third party's products or services or intellectual property in any way.

Except as provided in ARTERY's terms and conditions of sale for such products, ARTERY disclaims any express or implied warranty, relating to use and/or sale of the products, including but not restricted to liability or warranties relating to merchantability, fitness for a particular purpose (based on the corresponding legal situation in any unjudicial districts), or infringement of any patent, copyright, or other intellectual property right.

ARTERY's products are not designed for the following purposes, and thus not intended for the following uses: (A) Applications that have specific requirements on safety, for example: life-support applications, active implant devices, or systems that have specific requirements on product function safety; (B) Aviation applications; (C) Aerospace applications or environment; (D) Weapons, and/or (E) Other applications that may cause injuries, deaths or property damages. Since ARTERY products are not intended for the above-mentioned purposes, if purchasers apply ARTERY products to these purposes, purchasers are solely responsible for any consequences or risks caused, even if any written notice is sent to ARTERY by purchasers; in addition, purchasers are solely responsible for the compliance with all statutory and regulatory requirements regarding these uses.

Any inconsistency of the sold ARTERY products with the statement and/or technical features specification described in this document will immediately cause the invalidity of any warranty granted by ARTERY products or services stated in this document by ARTERY, and ARTERY disclaims any responsibility in any form.