

AMKN 系列工控板 网络通信说明

(2026 年 8 月 3 日修订版)

版权声明

本产品使用手册包含的所有内容均受版权法的保护，未经北京中嵌凌云电子有限公司的书面授权，任何组织和个人不得以任何形式或手段对整个手册和部分内容进行复制和转载。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我司概不承担其他责任。并且我司对本产品的销售和使用不作任何明示或暗示的担保，包括对产品特定用途的适用性，适销性或对任何专利权、版权或其他知识产权的侵权责任等均不作担保。我司对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，我司可能随时会对产品描述和相关的功能调整或技术改进，保留修改文档中任何内容的权利，恕不另行通知。

商标声明

AMKN 系北京中嵌凌云电子有限公司注册商标，未经书面授权，任何人不得以任何方式使用该商标、标记。

销售及服务网络

北京

销售电话：188 0108 0298

地址：北京市海淀区吴家场路 1 号院 2 号楼

邮箱：SALES@EMBEDARM.COM

西安

销售电话：189 9285 2102

地址：西安市曲江新区旺座曲江 H 座 3003 室

邮箱：SALES@EMBEDARM.COM

技术支持：

电话：029-8877 2044（工作日）

手机/微信：133 9928 8868

邮箱：EMBEDARM@126.COM

网址：WWW.ZQLYMCU.COM

版本

表格显示本产品使用手册在不同时期的修订版本及修订原因说明：

版本	修改内容	完成日期	修订部门
V1.00	正式发布	2026. 8. 3	研发部

适用产品型号：

序号	类型	订货型号	备注
1	工业控制模块	STM32F103VE、STM32F103ZE、STM32F107VC、STM32F407VE、STM32F407ZE、GD32F303VE、GD32F303ZE、GD32F307VE、STM32H723ZG、STM32H743XI、STM32MP157	
2	工业控制板	AMKN8602、AMKN8602G、AMKN8626、AMKN8628、AMKN8629、AMKN8630、AMKN8632、AMKN8638、AMKN8639、AMKN8701G、AMKN8702G、AMKN8703、AMKN8804、RTU-BUS、RTU-6XXX系列	

特别提醒：本软件只适用于以上产品硬件，并不适用于其它产品。

目录

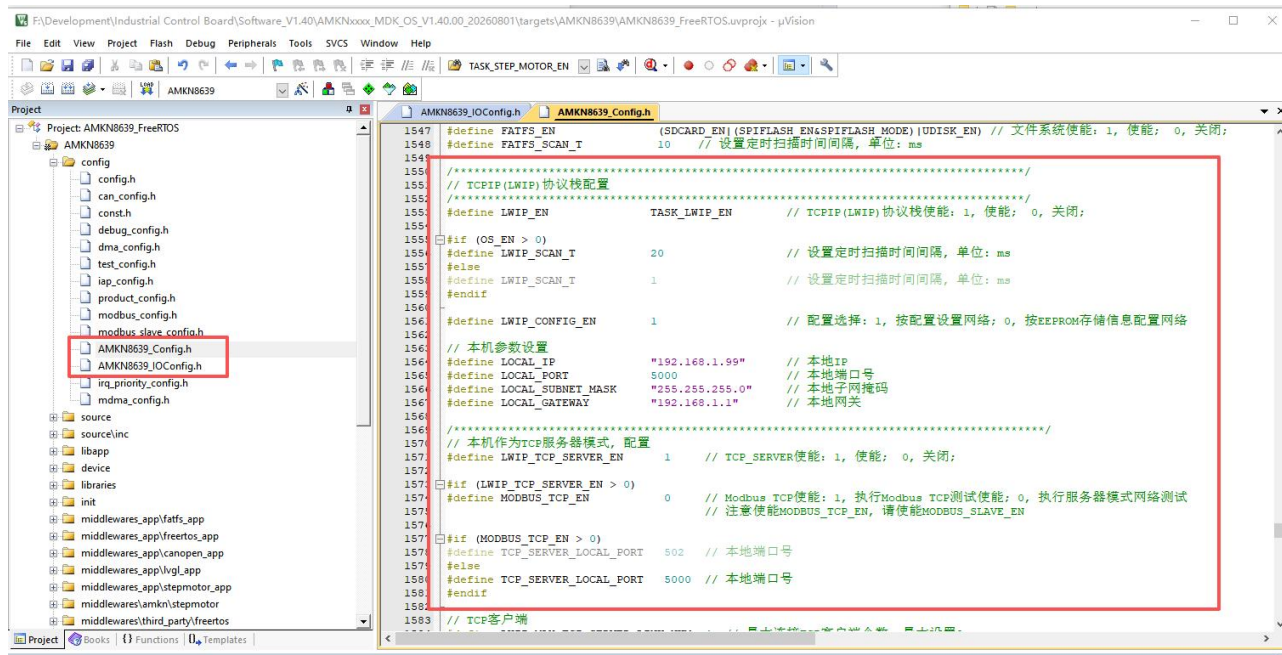
1. 前提说明	5
2. 打开软件	5
3. 关于 LWIP 的一些源程序说明	6
4. TCP 服务器通信模式	7
5. UDP 服务器通信模式	9
6. TCP 客户端通信模式	11
7. UDP 客户端通信模式	13
8. TFTP 服务器通信模式	15
附录：文档修改记录	18

1. 前提说明:

以下所述都以 AMKNxxxx_MDK_OS_V1.40.00_20260801 为参考, 选择 FreeRTOS 为操作系统, 以 AMKN8639 工控板为例说明

2. 打开软件

\targets\AMKN8639\AMKN8639_FreeRTOS.uvprojx, 如下图:



打开这 2 个红框内头文件:

AMKN8639_IOConfig.h 是所有硬件端口定义, 包括网络端口定义。

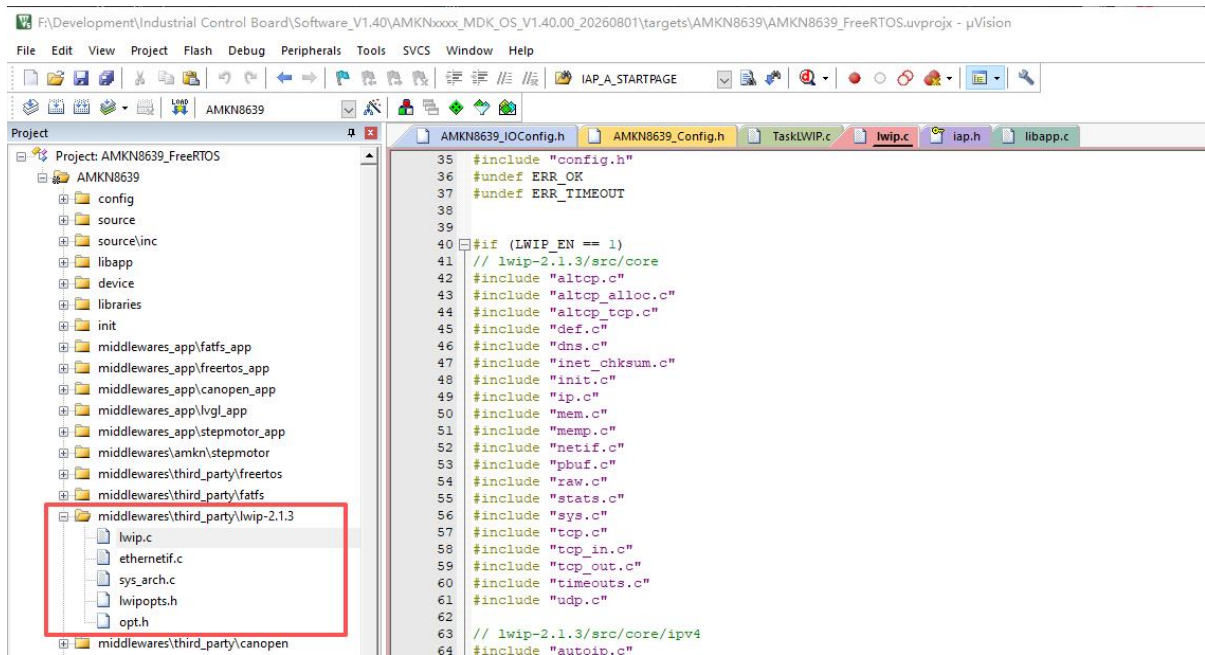
AMKN8639_Config.h 是所有功能定义, 用户可以查看 LWIP 网络部分功能定义, 一些重要配置如下:

- (1) LWIP_EN: LWIP 工作使能由 TASK_LWIP_EN 决定, 在文件开头定义
- (2) LWIP_SCAN_T: 扫描时间, 有操作系统一般定义 20ms, 无操作系统一般定义 1ms
- (3) LWIP_CONFIG_EN: 参数配置选择使能, 0, 按 EEPROM 存储信息配置网络;
1, 按以下内容配置网络;
- (4) LOCAL_IP: 本地 IP 地址, 默认 192.168.1.99
- (5) LOCAL_PORT: 本地端口号, 默认 5000
- (6) LOCAL_SUBNET_MASK: 本地子网掩码, 默认 255.255.255.0
- (7) LOCAL_GATEWAY: 本地网关, 默认 192.168.1.1
- (8) ETH_RXBUFNB: 驱动中接收缓存包数, 一般设置 3, 如果 SRAM 比较小, 也可以设置 2
- (9) ETH_MAX_RX_PACKET_SIZE: 驱动中接收缓存包大小, 固定 1520
- (10) ETH_TXBUFNB: 驱动中发送缓存包数, 一般设置 3, 如果 SRAM 比较小, 也可以设置 2
- (11) ETH_MAX_TX_PACKET_SIZE: 驱动中发送缓存包大小, 固定 1520

3. 关于 LWIP 的一些源程序说明

(1) lwip 代码源文件

在 middlewares/third_party/lwip/lwip-2.1.3 目录下：2.1.3 是 lwip 的版本号



lwip.c: 所有 lwip 的源码文件都包含在这个文件里;

ethernetif.c: 网络底层驱动接口文件

sys_arch.c: 操作系统接口文件

lwipopts.h: 主要 lwip 参数配置文件

opts.h: 全部 lwip 配置文件

(2) ./labapp/net_app.c 网络应用初始化, 里面重要函数介绍如下

NET_AppInit() 网络硬件初始化: 调用路径: main()->API_Init->NET_AppInit()

NET_AppProc() 网络应用处理函数, 主要功能初始化 lwip, 监控网络连接状态, 调用路径:

App_TaskInputData()->APP_InputData->NET_AppProc()

LwIP_Init: lwip 初始化函数, 调用路径:

App_TaskInputData()->APP_InputData->NET_AppProc()->LwIP_Init()

(3) ./source/TaskLWIP.c 网络各种通信模式下的任务, 里面重要函数介绍如下

LWIP_TaskCreate(): lwip 所有任务创建函数, 调用路径:

App_TaskInputData()->APP_InputData->NET_AppProc()->LWIP_TaskCreate()

App_TaskTCPCliet1-App_TaskTCPCliet4: TCP 客户端通信任务

App_TaskUDPCliet: UDP 客户端通信任务

App_TaskTCPSever: TCP 服务端通信任务

App_TaskUDPSever: UDP 服务端通信任务

App_TaskTFTPSever: TFTP 服务端通信任务

4. TCP 服务器通信模式

- (1) 模式配置：在 ./targets/AMKN8639/config/AMKN8639_Config.h 中：

将 LWIP_TCP_SERVER_EN 设置为 1，使能本模式；

TCP_SERVER_LOCAL_PORT 是 TCPServer 服务器端口号，默认 5000；在使能 MODBUS_TCP_EN 时默认是 502；

LWIP_MAX_TCP_SERVER_LINK_NUM 是可接入客户端最大数量，不可设置太大，一般 4 个就可以了。

- (2) 打开 ./source/TaskLWIP.c 中的函数 App_TaskTCPServer 是 TCPServer 任务，完成 TCP 服务器通信所有功能。

创建一个新的连接：netconn_new()

绑定本地端口号：netconn_bind()

连接进入监听模式：netconn_listen()

客户端连接：通过 netconn_accept() 函数实时监测是否有新的客户端连接，如果有加入客户端列表并调用 NET_TCPServerDataProcess 函数。

接收客户端数据：通过 netconn_recv() 函数接收客户端数据，如果收到数据并调用 NET_TCPServerDataProcess 函数；如果出现非超时错误也会调用 NET_TCPServerDataProcess 函数；

发送数据：调用函数 xQueueReceive 接收到用户要发送的数据指针和长度，再通过 netconn_write 函数发送出去。

- (4) 打开 ./source/user_netwifiapp.c 中的函数 NET_TCPServerDataProcess：

在该函数里用户可以记录客户的连接断开状态，并处理接收到的数据，还可以利用回调函数发送数据。

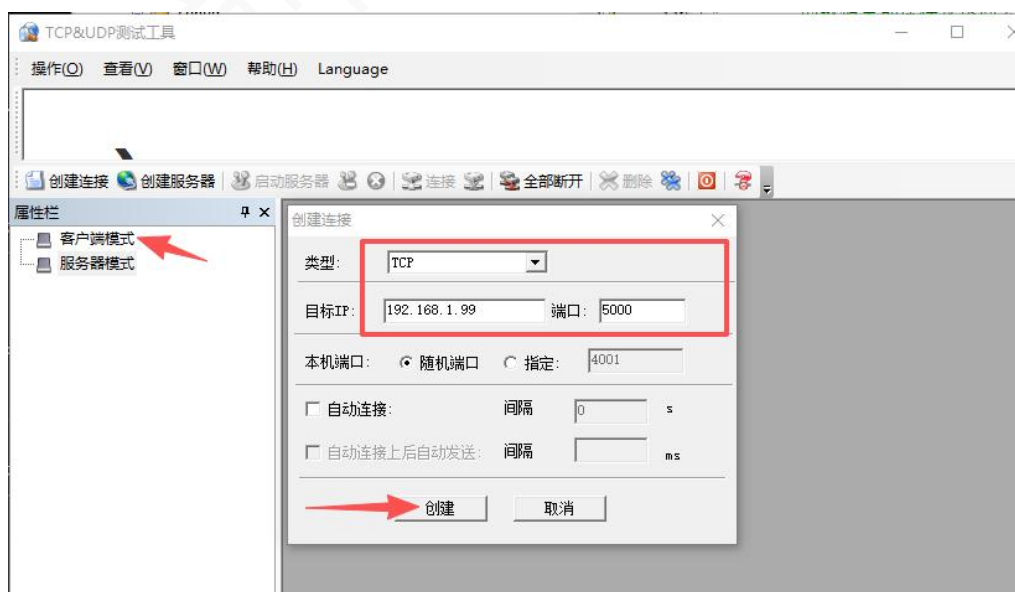
- (5) 打开 ./config/test_config.h，使能 APP_NET_TEST_EN 和 APP_NET_TCP_SERVER_TX_TEST_EN。

在 ./source/user_testapp.c 中查看 NET_UserTCPServer_TxTest 函数：判断是否有客户端连接，有间隔 1 秒调用 NET_UserSendData 函数发送数据 “0123456789ABCDEF\r\n”。调用路径：

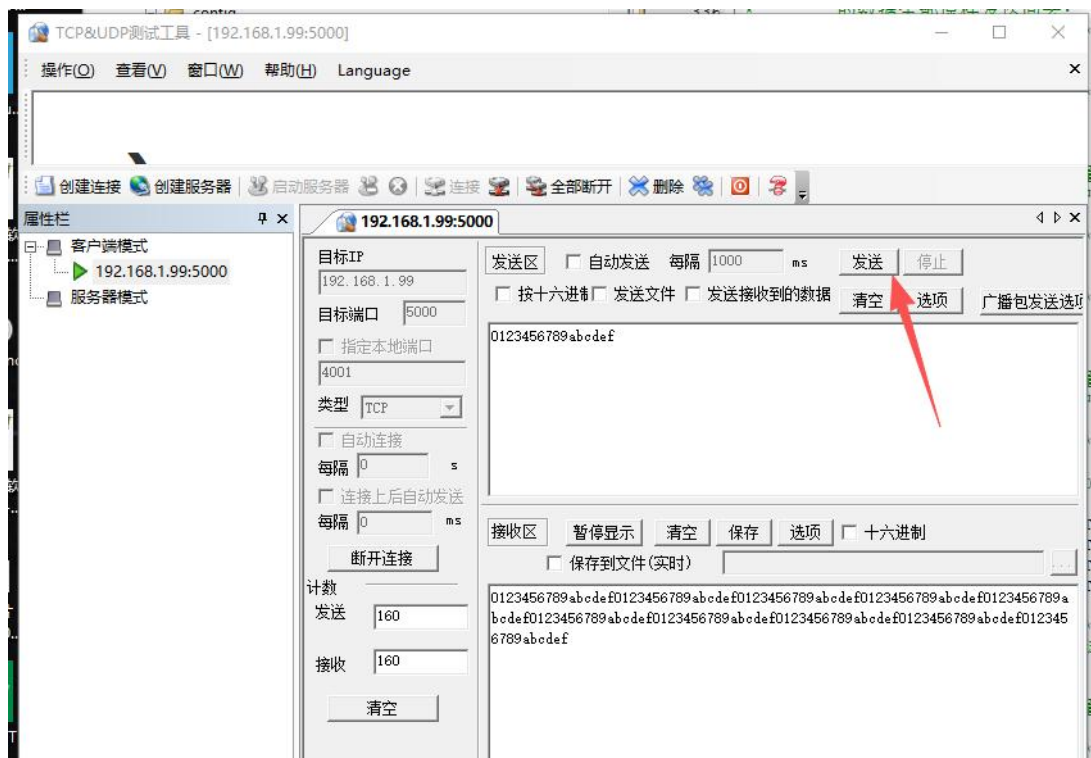
App_TaskTest->User_AppTest->NET_UserTCPServer_TxTest()->NET_UserSendData()

- (6) 用网线连接板子网口和计算机，打开网络测试软件，如图：

创建客户端

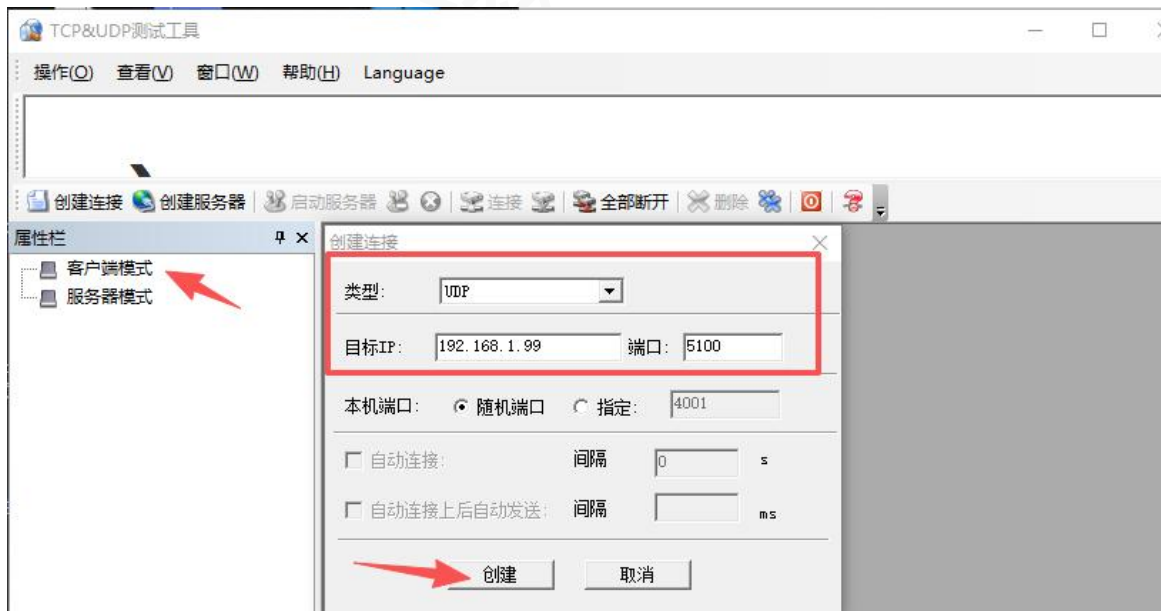


连接服务器，发送数据测试:发送“0123456789abcdef”后，板子同样返回数据“0123456789abcdef”

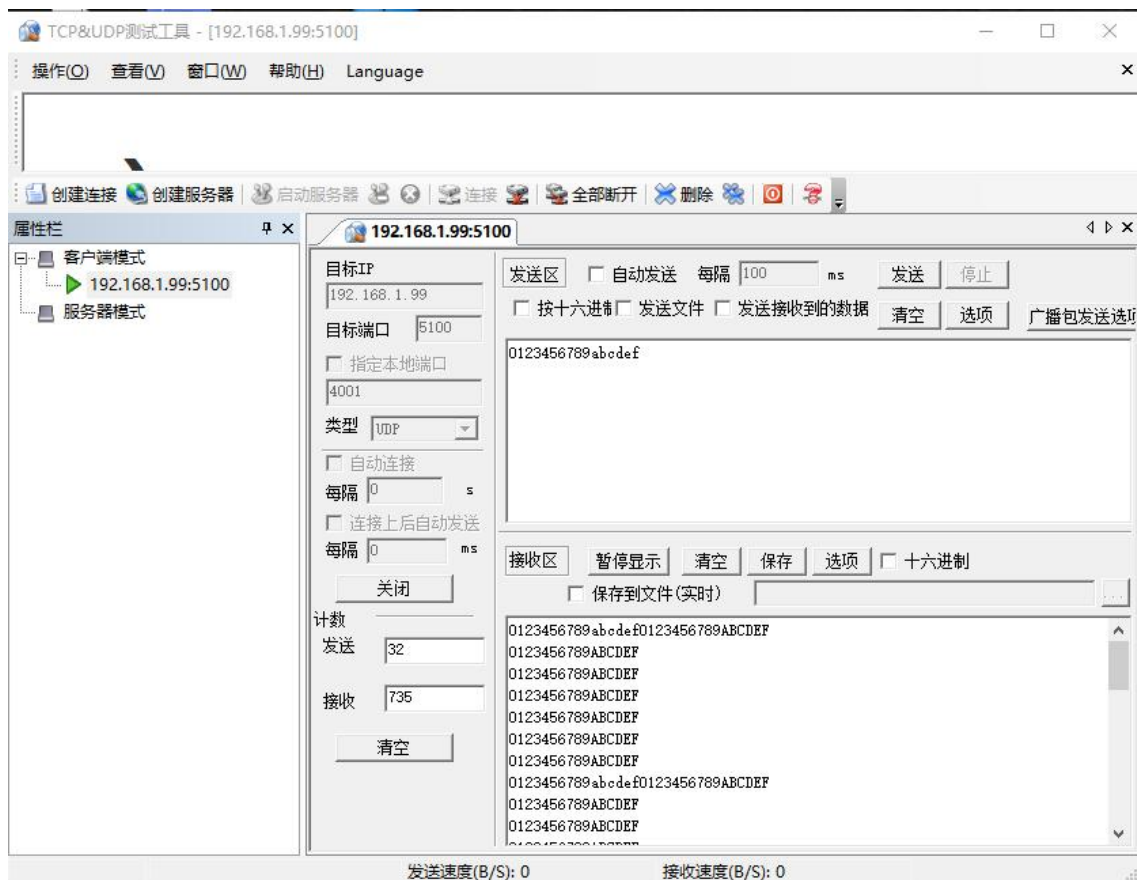


5. UDP 服务器通信模式

- (1) 模式配置：在 `./targets/AMKN8639/config/AMKN8639_Config.h` 中：
将 `LWIP_UDP_SERVER_EN` 设置为 1，使能本模式；
`UDP_SERVER_LOCAL_PORT` 是 UDP 服务器端口号，默认 5100；
`LWIP_MAX_UDP_SERVER_LINK_NUM` 是可接入客户端最大数量，不可设置太大，一般 4 个就可以了。
- (2) 打开 `./source/TaskLWIP.c` 中的函数 `App_TaskUDPServer` 是 UDP 服务器任务，完成 UDP 服务器通信所有功能。
创建一个新的连接：`netconn_new()`
绑定本地端口号：`netconn_bind()`
接收客户端数据：通过 `netconn_recv()` 函数接收客户端数据，如果收到数据并调用 `NET_UDPServerDataProcess` 函数；如果出现非超时错误也会调用 `NET_UDPServerDataProcess` 函数；
发送数据：调用函数 `xQueueReceive` 接收到用户要发送的数据指针和长度，再通过 `netconn_send` 函数发送出去。
- (3) 打开 `./source/user_netwifiapp.c` 中的函数 `NET_UDPServerDataProcess`：
在该函数里用户可以处理接收到的数据，还可以利用回调函数发送数据。
- (4) 打开 `./config/test_config.h`，使能 `APP_NET_TEST_EN` 和 `APP_NET_UDP_SERVER_TX_TEST_EN`。
在 `./source/user_testapp.c` 中查看 `NET_UserUDPServer_TxTest` 函数：判断是否有客户端连接，有间隔 1 秒调用 `NET_UserSendData` 函数发送数据 “0123456789ABCDEF\r\n”。调用路径：
`App_TaskTest`→`User_AppTest`→`NET_UserUDPServer_TxTest()`→`NET_UserSendData()`
- (5) 用网线连接板子网口和计算机，打开网络测试软件，如图：
创建 UDP 客户端



连接服务器后，发送数据测试：发送“0123456789abcdef”后，板子同样返回数据“0123456789abcdef”，之后，板子间隔 1 秒发送数据“0123456789ABCDEF”



6. TCP 客户端通信模式

- (1) 模式配置：在 ./targets/AMKN8639/config/AMKN8639_Config.h 中：

将 LWIP_TCP_CLIENT_EN 设置为 1，使能本模式；

TCP_CLIENT_LOCAL_PORT 是本地客户端端口号，默认 5200；

LWIP_TCP_DSC1_EN-LWIP_TCP_DSC4_EN：远端服务器 1-4 使能，可以同时开启同时连接 4 个远程服务器，一般连接 1 个，使能 LWIP_TCP_DSC1_EN 即可。

LWIP_MAX_TCP_DSC_NUM 是远程服务器最大数量，不可设置太大，一般 2 个就可以了。

LWIP_TCP_DSC1_IP-LWIP_TCP_DSC4_IP：远程服务器的 IP 地址，客户根据实际需要修改；

LWIP_TCP_DSC1_PORT-LWIP_TCP_DSC4_PORT：远程服务器的端口号，客户根据实际需要修改；

- (2) 打开 ./source/TaskLWIP.c 中的函数 App_TaskTCPClient1-App_TaskTCPClient4 是 4 个 TCP 客户端任务，完成 TCP 客户端通信所有功能。

客户端初始化：TCPClient_Init()；在 TCPClient_Init() 函数当中：

创建一个新的连接：netconn_new()

绑定本地 IP 地址连接：netconn_bind()

客户端连接远端服务器：TCPClient_Connect()；在 TCPClient_Connect() 函数当中：

连接服务器：netconn_connect()，成功调用 NET_TCPClientDataProcess() 函数

客户端进入通信模式：TCPClient_IdleProc()；在 TCPClient_IdleProc() 函数当中：

接收服务器数据：通过 netconn_recv() 函数接收服务器数据，如果收到数据并调用

NET_TCPClientDataProcess 函数；如果出现非超时错误也会调用 NET_TCPClientDataProcess 函数；

发送数据：调用函数 xQueueReceive 接收到用户要发送的数据指针和长度，再通过 netconn_write 函数发送出去。

- (3) 打开 ./source/user_netwifiapp.c 中的函数 NET_TCPClientDataProcess：

在该函数里用户可以记录连接远端服务器状态，并处理接收到的数据，还可以利用回调函数发送数据。

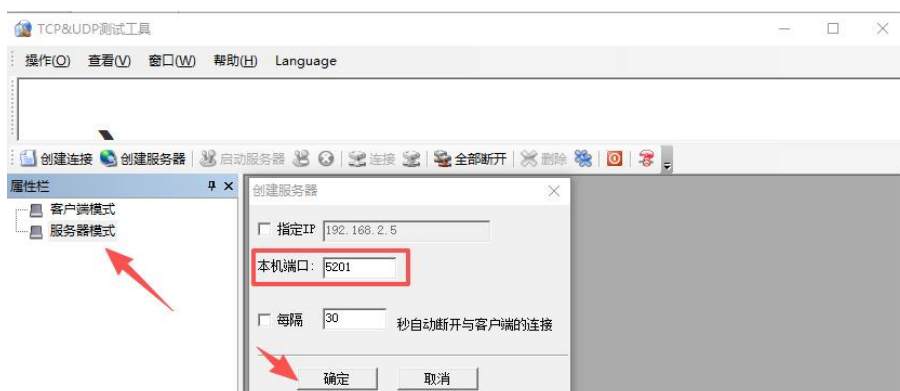
- (4) 打开 ./config/test_config.h，使能 APP_NET_TEST_EN 和 APP_NET_TCP_CLIENT_TX_TEST_EN。

在 ./source/user_testapp.c 中查看 NET_UserTCPClient_TxTest 函数：判断是否连接远端服务器成功后，间隔 1 秒调用 NET_UserSendData 函数发送数据 “0123456789ABCDEF\r\n”。

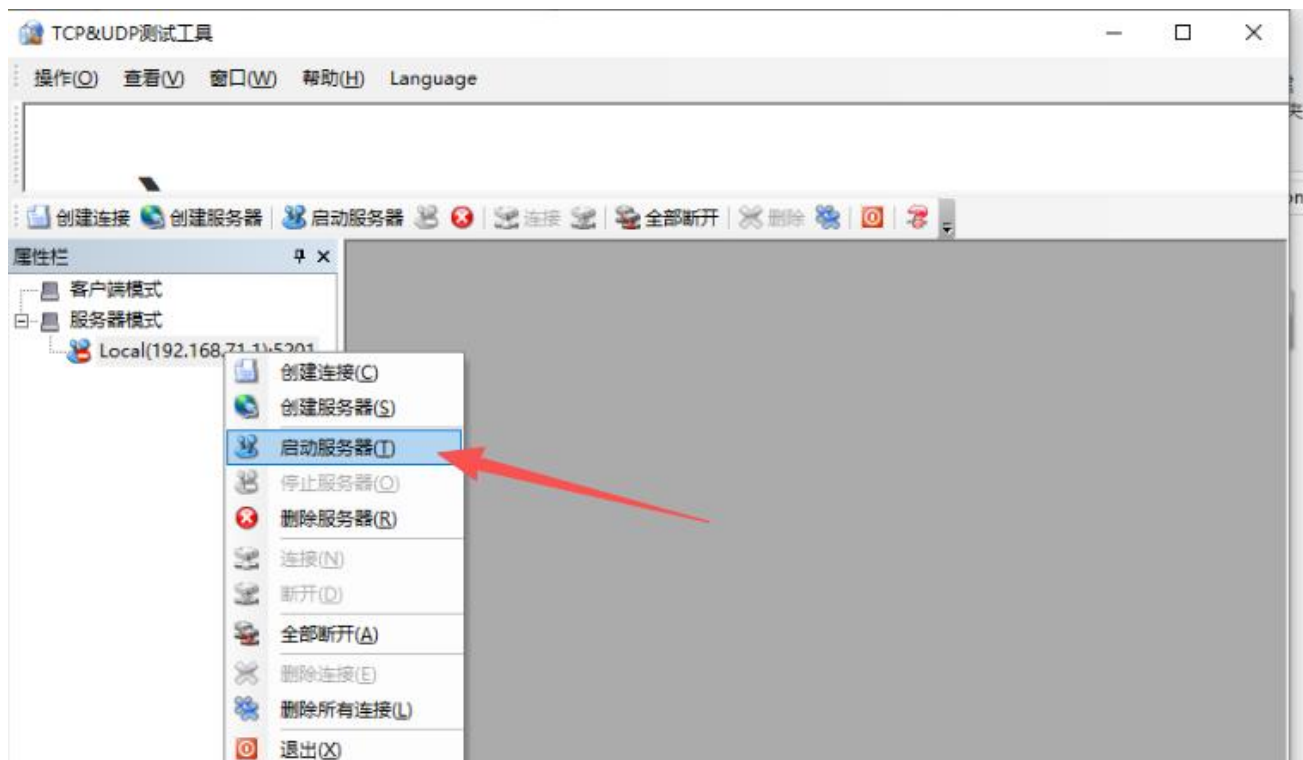
调用路径：

App_TaskTest->User_AppTest->NET_UserTCPClient_TxTest()->NET_UserSendData()

- (5) 用网线连接板子网口和计算机，打开网络测试软件，如图：创建服务器

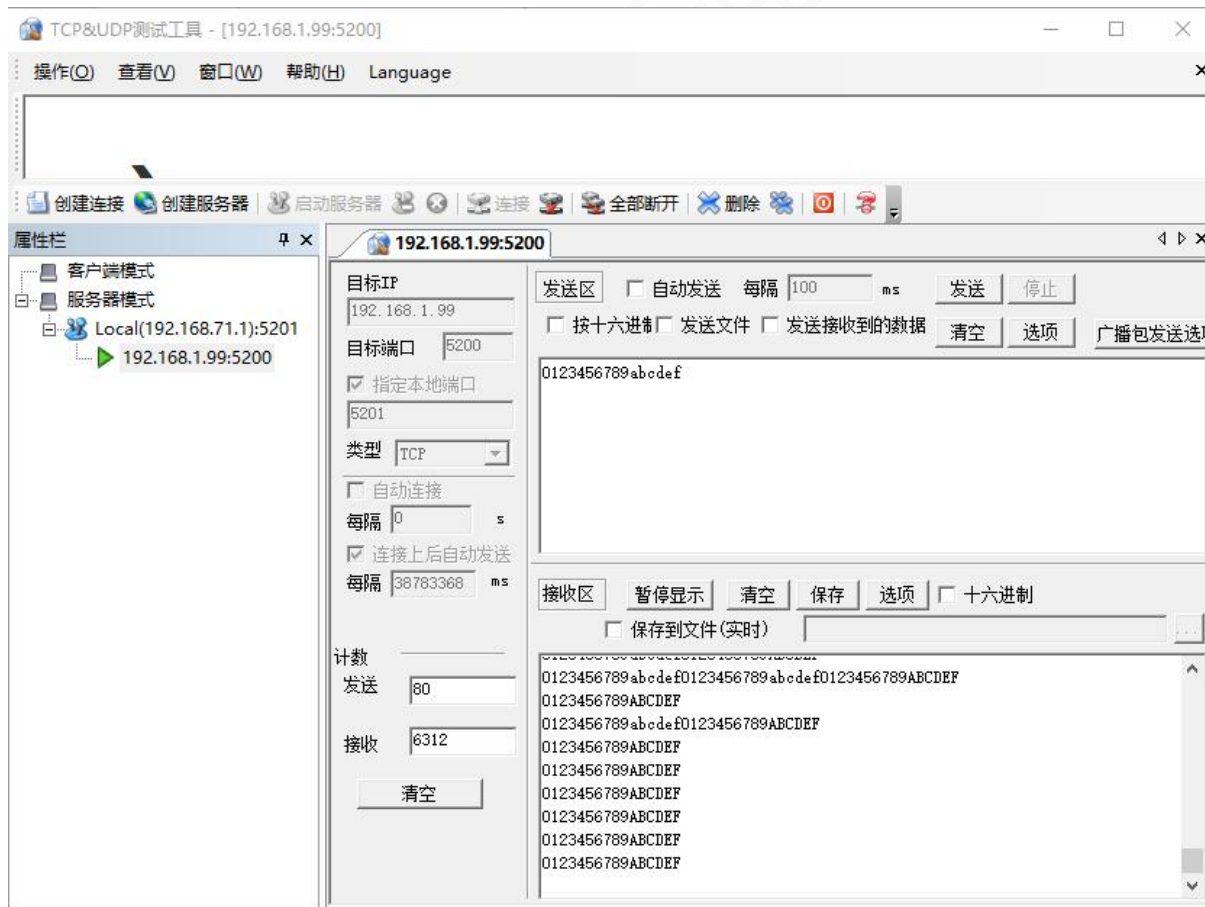


启动服务器：



板子启动后，连接服务器后主动间隔 1 秒发送“0123456789ABCDEF”；

发送数据测试:发送“0123456789abcdef”后，板子同样返回数据“0123456789abcdef”



7. UDP 客户端通信模式

- (1) 模式配置：在./targets/AMKN8639/config/AMKN8639_Config.h 中：

将 LWIP_UDP_CLIENT_EN 设置为 1，使能本模式；

UDP_CLIENT_LOCAL_PORT 是本地客户端端口号，默认 5300；

LWIP_MAX_UDP_DSC_NUM 是远程服务器最大数量，不可设置太大，一般 2 个就可以了。

LWIP_UDP_DSC1_IP-LWIP_UDP_DSC4_IP：远程服务器的 IP 地址，客户根据实际需要修改；

LWIP_UDP_DSC1_PORT-LWIP_UDP_DSC4_PORT：远程服务器的端口号，客户根据实际需要修改；

- (2) 打开./source/TaskLWIP.c 中的函数 App_TaskUDPCliant 是 UDP 客户端任务，完成 UDP 客户端通信所有功能。

创建一个新的连接：netconn_new()

绑定本地 IP 地址连接：netconn_bind()

接收服务器数据：通过 netconn_recv() 函数接收服务器数据，如果收到数据并调用 NET_UDPCliantDataProcess 函数；

发送数据：调用函数 xQueueReceive 接收到用户要发送的数据指针和长度，再通过 netconn_send 函数发送出去。

- (3) 打开./source/user_netwifiapp.c 中的函数 NET_UDPCliantDataProcess：

在该函数里处理接收到的数据，还可以利用回调函数发送数据。

- (4) 打开./config/test_config.h，使能 APP_NET_TEST_EN 和 APP_NET_UDP_CLIENT_TX_TEST_EN。

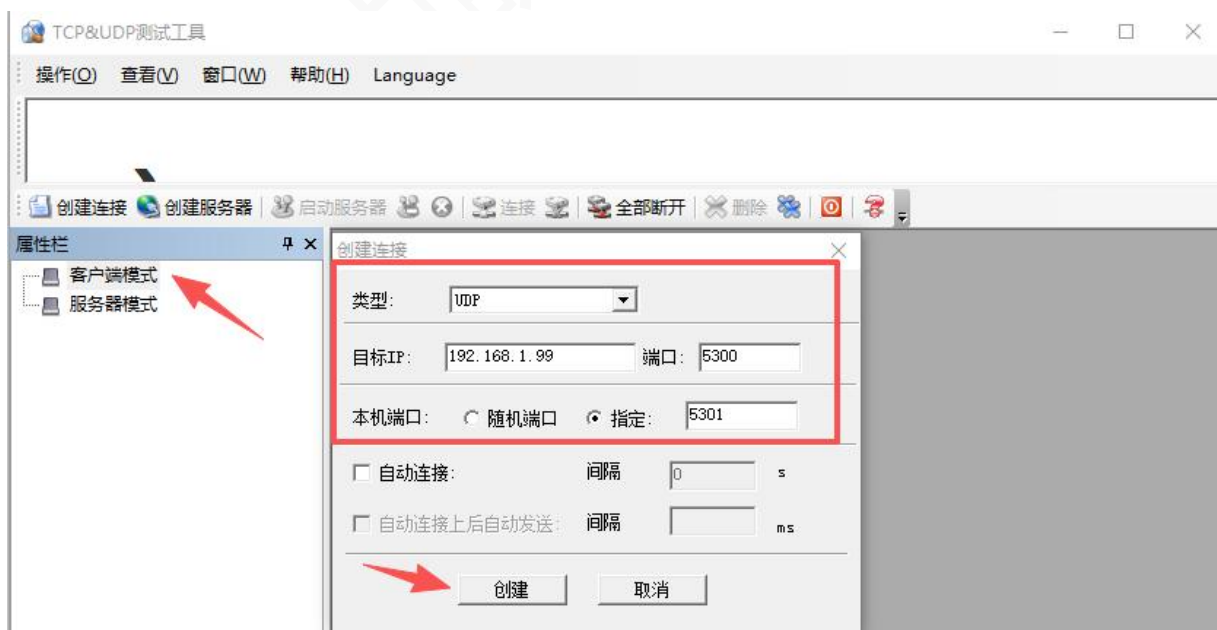
在./source/user_testapp.c 中查看 NET_UserUDPCliant_TxTest 函数：判断是否连接远端服务器成功后，间隔 1 秒调用 NET_UserSendData 函数发送数据“0123456789ABCDEF\r\n”。

调用路径：

App_TaskTest->User_AppTest->NET_UserUDPCliant_TxTest()->NET_UserSendData()

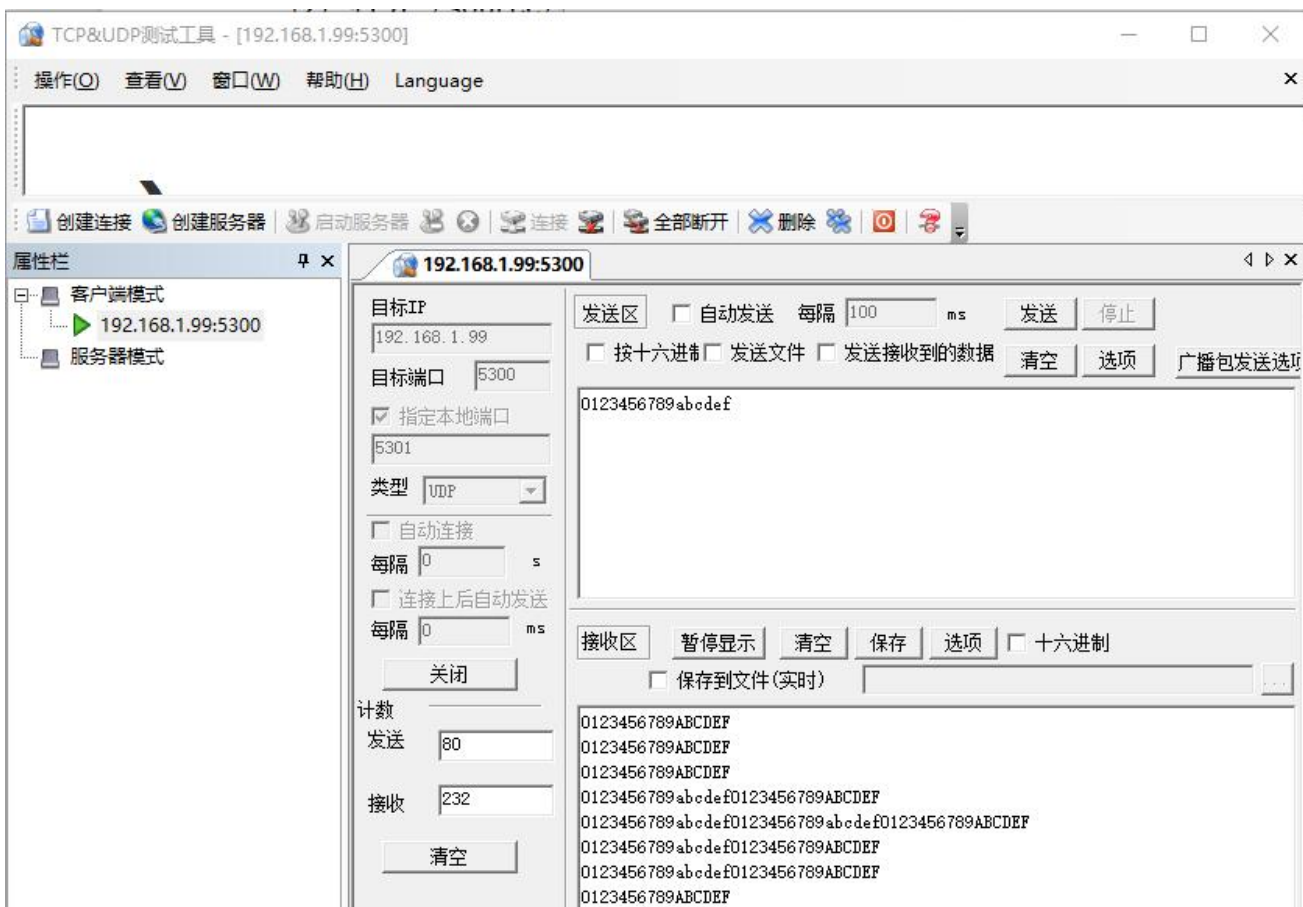
- (4) 用网线连接板子网口和计算机，打开网络测试软件，如图：

在客户端模式创建连接，注意：本地端口必须指定 5301，与 LWIP_UDP_DSC1_IP 保持一致



点击连接后，板子启动后主动间隔 1 秒发送 “0123456789ABCDEF”；

发送数据测试:发送 “0123456789abcdef” 后，板子同样返回数据 “0123456789abcdef”



8. TFTP 服务器通信模式

(1) 模式配置：在 ./targets/AMKN8639/config/AMKN8639_Config.h 中：

将 LWIP_TFTP_SERVER_EN 设置为 1，使能本模式；如果 IAP_EN 和 IAP_TFTP_EN 使能则自动使能 LWIP_TFTP_SERVER_EN；

TFTP_SERVER_LOCAL_PORT 是 TFTP 服务器端口号，固定是 69；

TFTP_FILE_DISK：选择文件存储介质：0，SPIFLASH_DISK；1，SD 卡；2，U 盘；3，NAND FLASH
默认选择 1

TFTP_RRQ_FILE_EN：设置为 1，利用 TFTP 读取文件使能；

TFTP_WRQ_FILE_EN：设置为 1，利用 TFTP 写入文件使能；

TFTP_WEQ_FILE_MODE：0，如果文件存在，停止写，返回错误；

1，如果文件存在，打开文件，并在文件结尾添加数据；

2，如果文件存在，删除原有文件，新建文件；

LWIP_MAX_TFTP_SERVER_LINK_NUM 是可接入客户端最大数量，不可设置太大，一般 1 个就可以了。

(2) 打开 ./source/TaskLWIP.c 中的函数 App_TaskTFTPServer 是 TFTP 服务器任务，完成 TFTP 通信所有功能。

TFTP 应用初始化函数：TFTP_AppInit()

创建一个新的连接：netconn_new()

绑定本地端口号：netconn_bind()

接收客户端数据：通过 netconn_recv() 函数接收客户端数据，如果收到数据并调用 TFTP_AppProc 函数进行处理；

函数；

发送数据：通过回调函数 TFTPServer_Write 发送数据。

(3) 打开 ./libapp/tftp.c 文件：

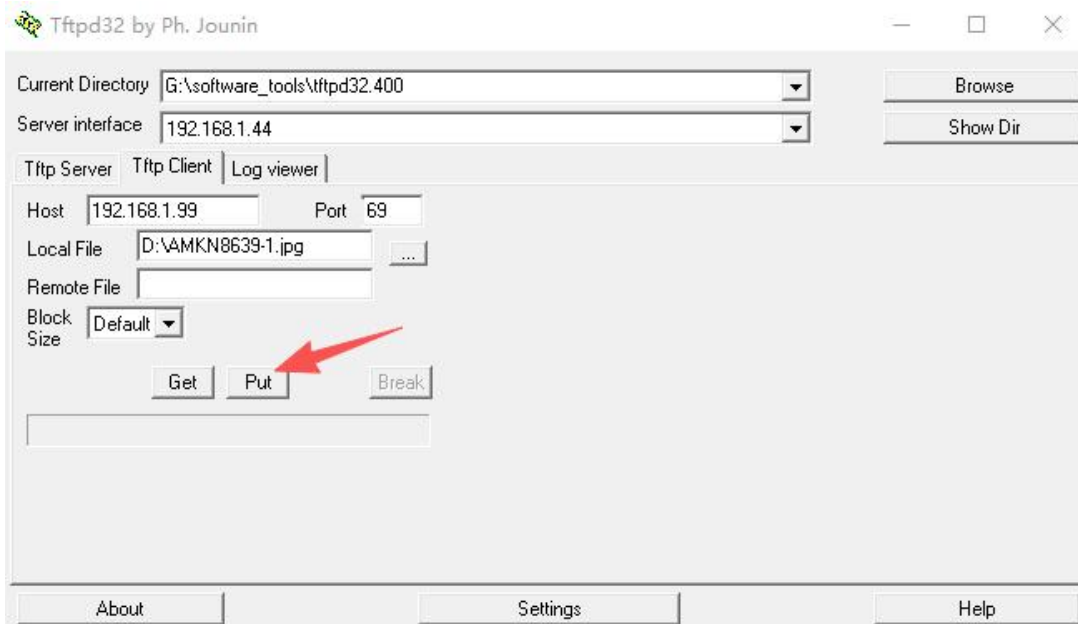
TFTP_AppInit：TFTP 应用初始化函数；

TFTP_AppProc：TFTP 应用处理函数，处理 TFTP 指令，读写文件，并通过回调函数发送数据；

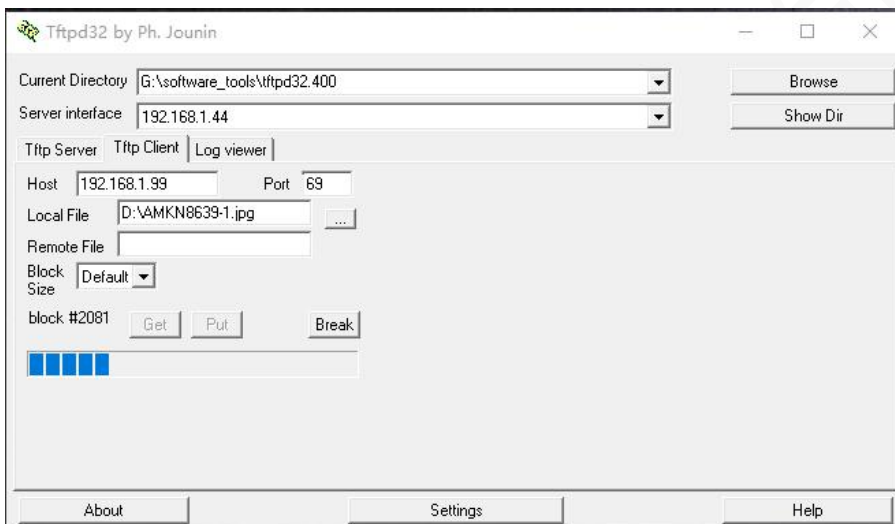
(4) 下载文件测试

用网线连接板子网口和计算机，打开 TFTP 测试软件，如图：

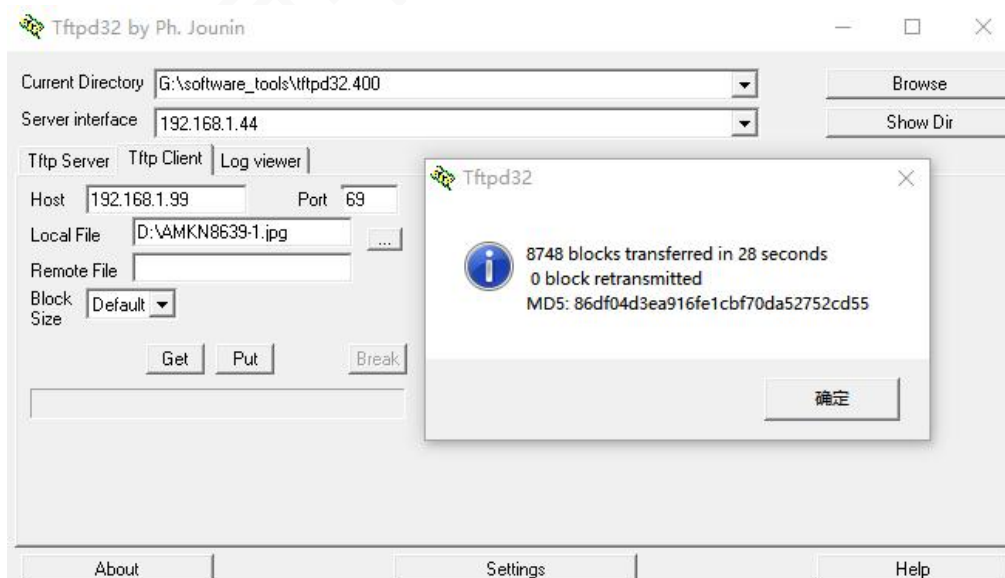
配置 TFTP 客户端：下载 D:\AMKN8639-1.jpg 图片文件到 AMKN8639 板子上的 SD 卡内



点击 Put 开始下载:



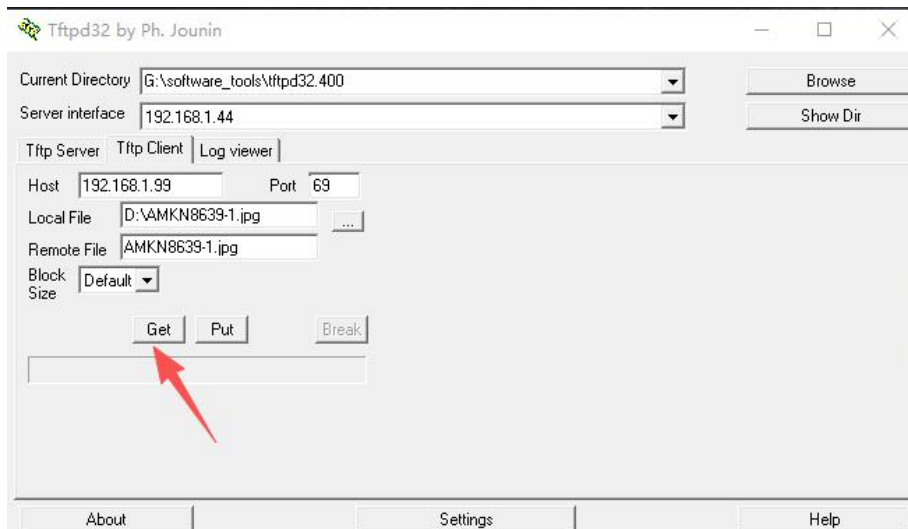
下载完成:



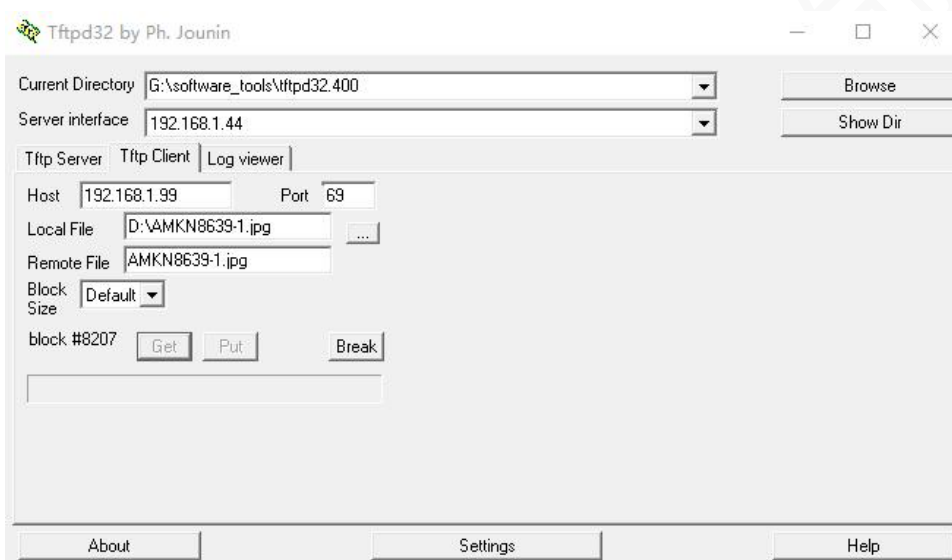
(5) 上传文件测试

将 D:\AMKN8639-1.jpg 图片文件改名字: AMKN8639-2.jpg

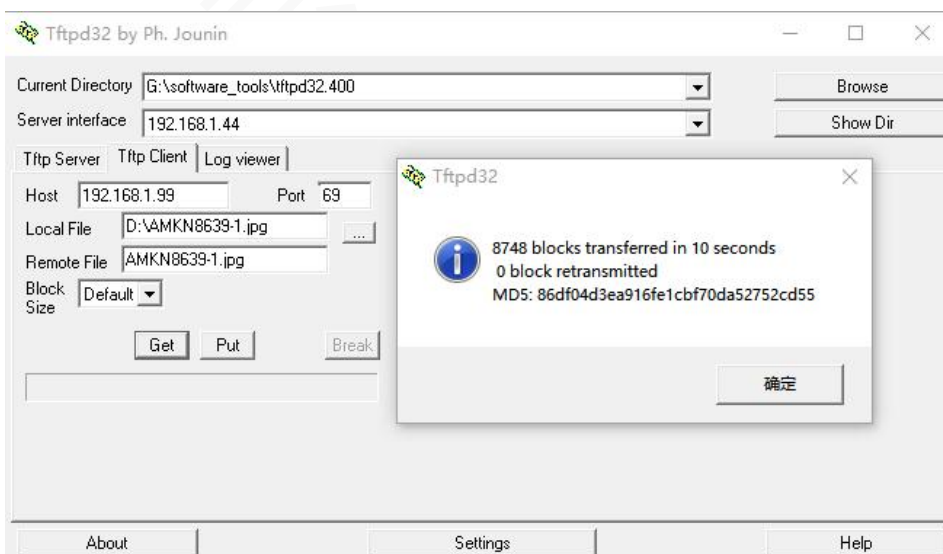
配置 TFTP 客户端: 上传 AMKN8639 板子上 SD 卡内文件 AMKN8639-1.jpg 文件到 D:\盘



点击 Get 开始上传: 注意无进度条只有 block 的数量显示



上传完成: 可以在 D:\下对比文件 AMKN8639-1.jpg 和 AMKN8639-2.jpg 是否正确



附录：文档修改记录

序号	版本	时间	更新内容
1	V1.00	2026.8.3	正式发布。