



欣瑞达串口屏以太网及 WiFi 使用教程

深圳市欣瑞达液晶显示技术有限公司



欣瑞达液晶



400-069-8808 0755-26018666



广东省深圳市南山区高新园北松坪山路5号嘉达研发大厦A座3层



www.xrd-lcd.com



使用淘宝扫一扫
欣瑞达官方旗舰店



使用微信扫一扫
欣瑞达官方微信

目录

1. 适用范围	2
2. 串口屏网络功能介绍	2
3. 如何使用 HTTP 协议与服务器进行数据交互和文件下载	2
3.1 搭建云端服务器	2
3.2 配置串口屏网络参数	2
3.2.1 工程中配置串口屏网络参数	3
3.2.2 使用脚本配置网络参数	5
3.3 串口屏与服务器数据交互	6
3.4 从云端更新某张图片或某个视频	7
4. 如何使用 FTP 协议远程更新串口屏工程和固件	8
4.1 搭建服务器	8
4.2 远程升级	9
5. 如何使用 TCP 协议进行 SOCKET 通信	9
5.1 NetAssist 配置	10
5.2 工程配置	11
5.3 修改服务器 IP、PORT	12
5.4 SOCKET 屏幕收	13
5.5 SOCKET 屏幕发	14

1. 适用范围

本文档适合欣瑞达所有带以太网或 WiFi 的串口屏产品。

2. 串口屏网络功能介绍

我司物联型串口屏引入了物联网的概念，通过屏上以太网接口或 WIFI 模块 可以实现将串口屏采集的数据上传到云端服务器；或者串口屏接收云端服务器上的信息，实现控制设备完成对应的操作。目前，带 以太网 或 WIFI 功能的串口屏常用于以下两种情形：

1. 使用 HTTP 协议与服务器进行数据交互和文件下载；
2. 使用 FTP 协议远程更新串口屏工程和固件；
3. TCP 通信。

3. 如何使用 HTTP 协议与服务器进行数据交互和文件下载

在我司提供参考资料中有关于云端服务器与串口屏通讯的示例工程，为了方便理解文档所提到的参数设置，在阅读文档的同时可以查看工程配置协助理解文档内容。实现云端服务器与串口屏网络通讯需要完成以下四个步骤：

1. 搭建云端服务器；（如果已有云端服务器可以跳过该步骤）
2. 配置串口屏网络参数；
3. 与服务器进行数据交互；
4. 使用 HTTP 协议下载文件。

3.1 搭建云端服务器

服务器，也称伺服器，是提供计算服务的设备。服务器的构成包括处理器、硬盘、内存、系统总线等，和通用的计算机架构类似，但是由于需要提供高可靠的服务，因此在处理能力、稳定性、可靠性、安全性、可扩展性、可管理性等方面要求较高。

搭建服务器这一步比较复杂，如果没有服务器建议寻找专业的开发服务器团队进行开发，以节省开发项目的时间。这一章节串口屏与云端服务器通讯的示例工程使用的服务器为公共服务器。

3.2 配置串口屏网络参数

此步骤为配置串口屏默认的网络参数，有两种实现方法：

1. 工程中配置串口屏网络参数；
2. 使用脚本配置网络参数。

3.2.1 工程中配置串口屏网络参数

1. 双击图 3-1 中【工程窗口】中的工程名字，右边的【属性窗口】会显示该工程的属性；

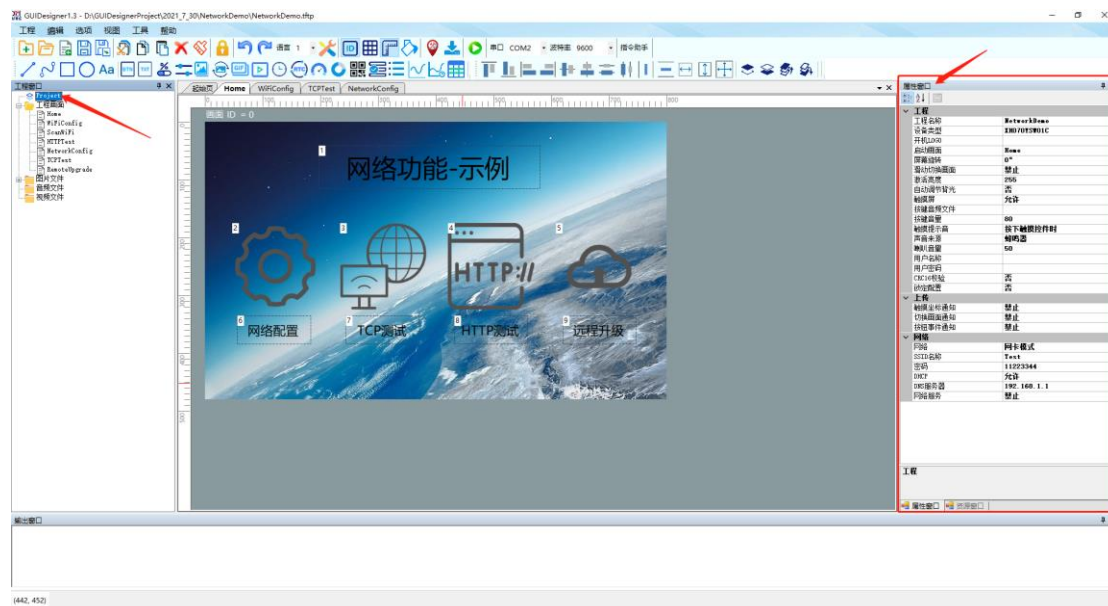


图 3-1 工程属性

2. 找到属性窗口中的网络设置，如图 3-2 所示；

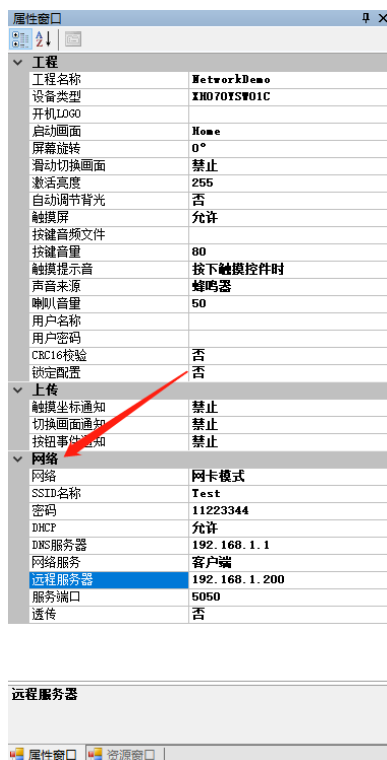


图 3-2 工程属性

3. 【网络】的参数设置:

网络: 该参数有 3 种模式: 禁用、网卡模式和热点模式; 此处选择“网卡模式”, 如图 3-3 所示;

- a) 禁用: 禁用网络功能;
- b) 网卡模式: 串口屏可以连接到以太网或 WiFi, 参数 SSID 名称和密码为串口屏上电后自动连接的 WiFi 设备名称和密码;
- c) 热点模式: 自建 WiFi 网络, 等待别的设备连接到自身, 如手机上的热点一样, 参数 SSID 名称和密码是为串口屏广播热点的名称和密码;

网络	网卡模式
SSID名称	禁止
密码	网卡模式
DHCP	热点模式
DNS服务器	192.168.1.1
网络服务	客户端
远程服务器	192.168.1.200
服务端口	5050
透传	否

图 3-3 网络选择

SSID 名称和密码: 选择“网卡模式”时, 填写的 SSID 名称和密码为串口屏上电后自动连接 WIFI 设备的 SSID 名称和密码; 选择“热点模式”时, 填写的 SSID 名称和密码为串口屏作为热点的名称和密码, 如图 3-4 所示;

网络	网卡模式
SSID名称	Test
密码	11223344

图 3-4 WiFi 名称和密码

DHCP: 动态获取设备的 IP 地址; 建议选择“允许”, 如图 3-5 所示;

- a) 允许: 串口屏使用动态 IP 地址, 直接由路由器地址分发 IP 等参数;
- b) 禁止: 串口屏使用静态 IP, 需要填写 IP 地址、子网掩码、默认网关、DNS 服务器等参数。

DHCP	允许
DNS服务器	禁止
网络服务	允许

图 3-5 DHCP

网络服务: 设置串口屏为客户端或服务器, 此处选择“客户端”, 如图 3-6 所示;

- 禁用 TCP 客户端、服务器通信功能；
- 客户端：串口屏作为客户端可连接到指定的服务器，需要填写远程服务地址（指定连接的服务器地址）、服务器端口号（由服务器决定）、网络透传选择否；
- 服务器：串口屏作为服务器等待客户端连接，需要填写服务器端口号、网络透传选择否。

网络服务	客户端
远程服务器	禁止
服务端口	客户端
透传	服务器

图 3-6 网络服务

远程服务器：选择“客户端”时，此处需要填写接入的服务器地址，如果选择“服务器”，此处不用填写，如图 3-7 所示。

服务器端口：服务器开放给客户端接入的通道。

网络透传：选择“是”后，服务器发送的数据到串口屏，串口屏会将同样的数据发送给 MCU。

网络服务	客户端
远程服务器	192.168.1.200
服务端口	5050
透传	否

图 3-7 网络服务

如工程中只使用 HTTP 协议和 FTP 协议与服务器通讯，只需要确保连接的路由器设备能够连接上网络，其它参数可以忽略。

3.2.2 使用脚本配置网络参数

打开工程所在的目录，打开 main.lua 文件添加修改默认网络参数的程序；下文 LUA 程序所使用的 API 接口函数参考《LUA 脚本 API》文档；配置网络参数的实现过程可以参考示例工程的 NetworkConfig 画面，如图 3-8 所示；点击“保存设置”按钮调用 LUA 脚本中的 set_network_cfg() 函数，然后调用 save_network_cfg() 保存网络设置，LUA 程序如下：

```

107
108
109
110
111
112
113
114
local dhcp
if get_text(screen, 4) == "禁用" then
    dhcp = 0
else
    dhcp = 1
end
set_network_cfg(dhcp, get_text(screen, 5), get_text(screen, 6), get_text(screen, 7), get_text(screen, 8))
save_network_cfg()
    
```

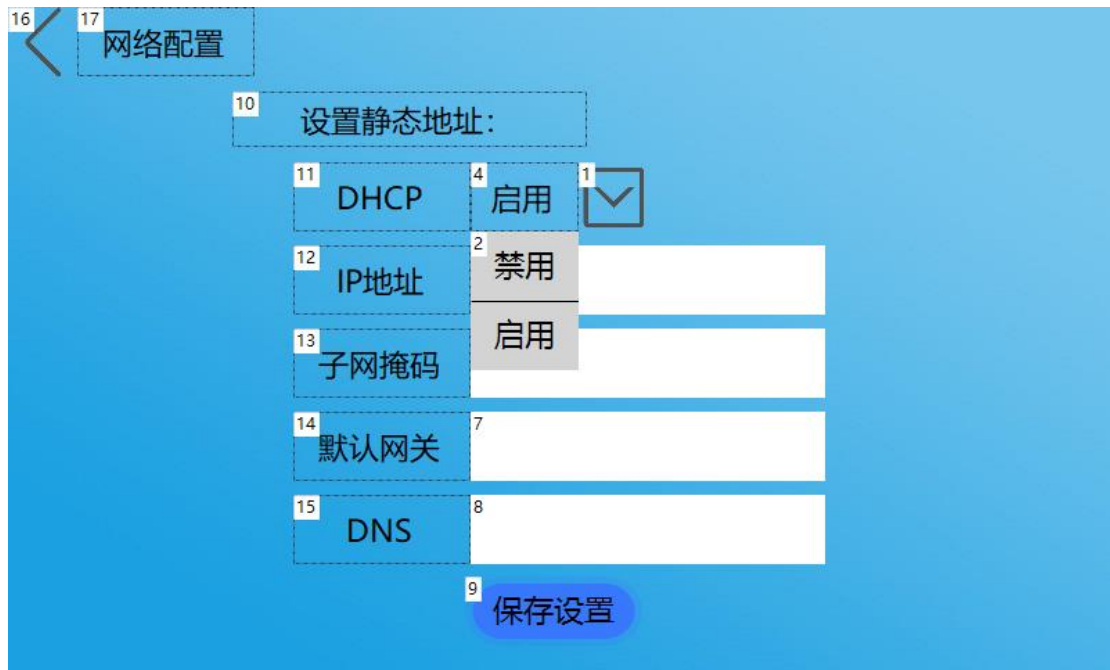


图 3-8 配置网络参数画面

3.3 串口屏与服务器数据交互

示例工程是使用 HTTP 协议与公共服务器进行数据交互，工程中与服务器通信的画面配置如图 3-9 所示；点击“HTTP-GET”按钮调用 API 函数 `http_request()` 请求服务器信息，接收到服务器返回的信息后串口屏会响应调用 `on_http_response()` 处理服务器返回的信息，工程请求的服务器信息和处理返回服务器信息的程序如下：

```
84      --获取公网入口IP地址
85      http_request(1000, "http://pv.sohu.com/cityjson?ie=utf-8", 0)
86      set_text(sc_HTTPTest, 2, "wait http response...")
87
161  --系统回调http响应,
162  function on_http_response(taskid, resp)
163      if taskid == 1000 then
164          set_text(sc_HTTPTest, 2, resp) --响应信息在文本控件上显示
165      end
166  end
```

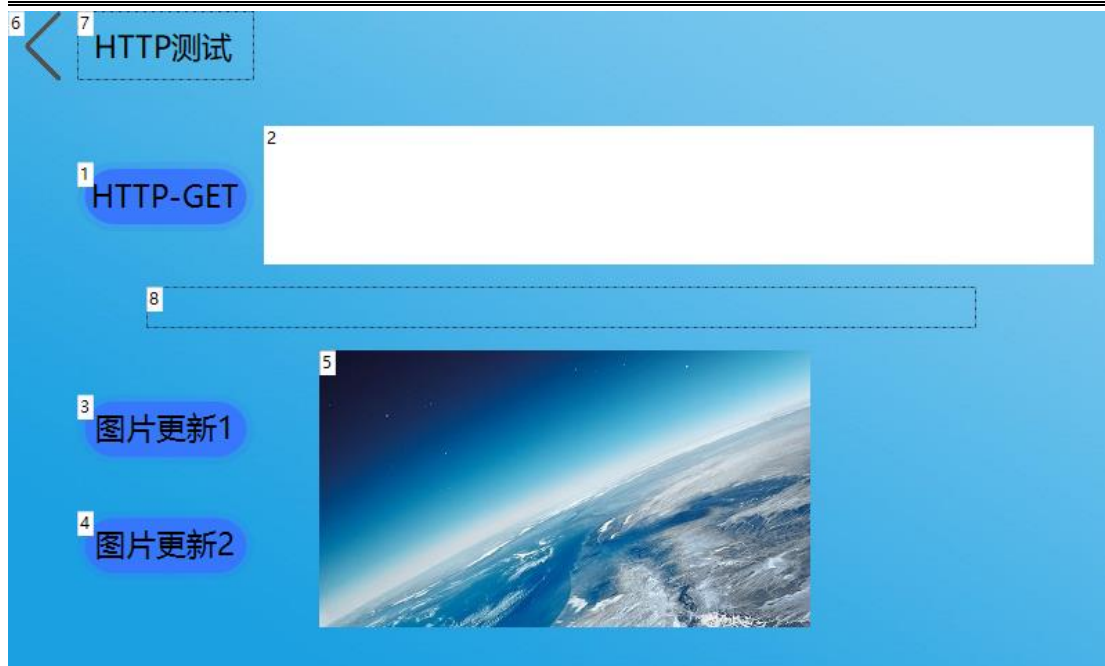


图 3-9 HTTP 测试页面

3.4 从云端更新某张图片或某个视频

串口屏提供使用 HTTP 协议的 API 函数，调用 HTTP 协议下载文件的 API 函数 可以实现远程更新某张图片、某个视频或者下载某个固件包到串口屏中。

下载文件的实现过程可以参考示例工程的 HTTPTest 画面，如图 3-9 所示；点击“图片更新 1”或“图片更新 2”按钮调用 LUA 脚本中 http_download() 函数，下载完成后串口屏会调用 on_http_download() 反馈下载信息，示例工程中 LUA 程序如下：

```
88      --使用HTTP协议从服务器下载文件
89      elseif control == 3 then
90          -- http_download(1, "此处填写HTTP图片文件地址", g_download_dir.."pic1.jpg")
91          set_text(sc_HTTPTest, 8, "图片1下载中...")
92          if g_surface ~= 0 then
93              destroy_surface(g_surface)
94              g_surface = 0
95          end
96
97      elseif control == 4 then
98          -- http_download(2, "此处填写HTTP图片文件地址", g_download_dir.."pic2.jpg")
99          set_text(sc_HTTPTest, 8, "图片2下载中...")
100         if g_surface ~= 0 then
101             destroy_surface(g_surface)
102             g_surface = 0
103         end
104     end
end
```


4. 如何使用 FTP 协议远程更新串口屏工程和固件

串口屏支持使用 FTP 协议进行在线更新工程和固件。通过 FTP 服务器将 PKG 文件下载到串口屏后，串口屏会自动解压更新压缩包里的内容。具体的实现步骤如下：

1. 将 PKG 文件放入 FTP 服务器中（如果没有服务器则需要自己搭建）；
2. 调用 API 接口函数从 FTP 服务器下载 PKG 文件。

4.1 搭建服务器

如有服务器则不需要进行此步骤。

本示例所使用的 FTP 服务器是使用免费开源软件 FileZillaFTP 搭建，如图 4-1 所示；安装完此软件，然后运行 FileZilla ServerInterface.exe 进行 FTP 用户和权限配置，如图 4-2 所示。

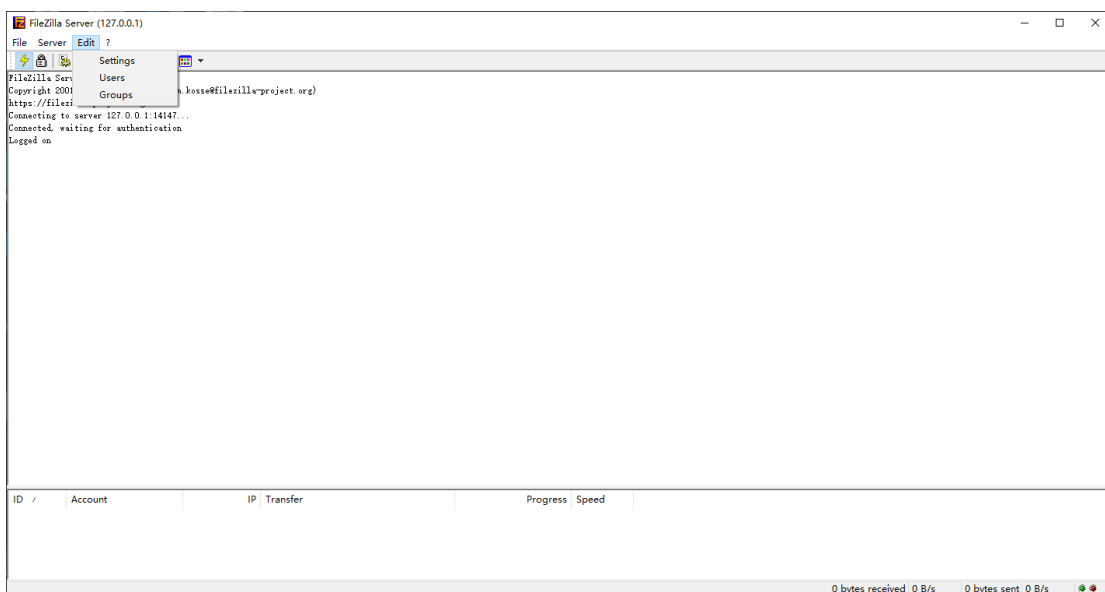


图 4-1 设置搭建 FTP 本地服务器

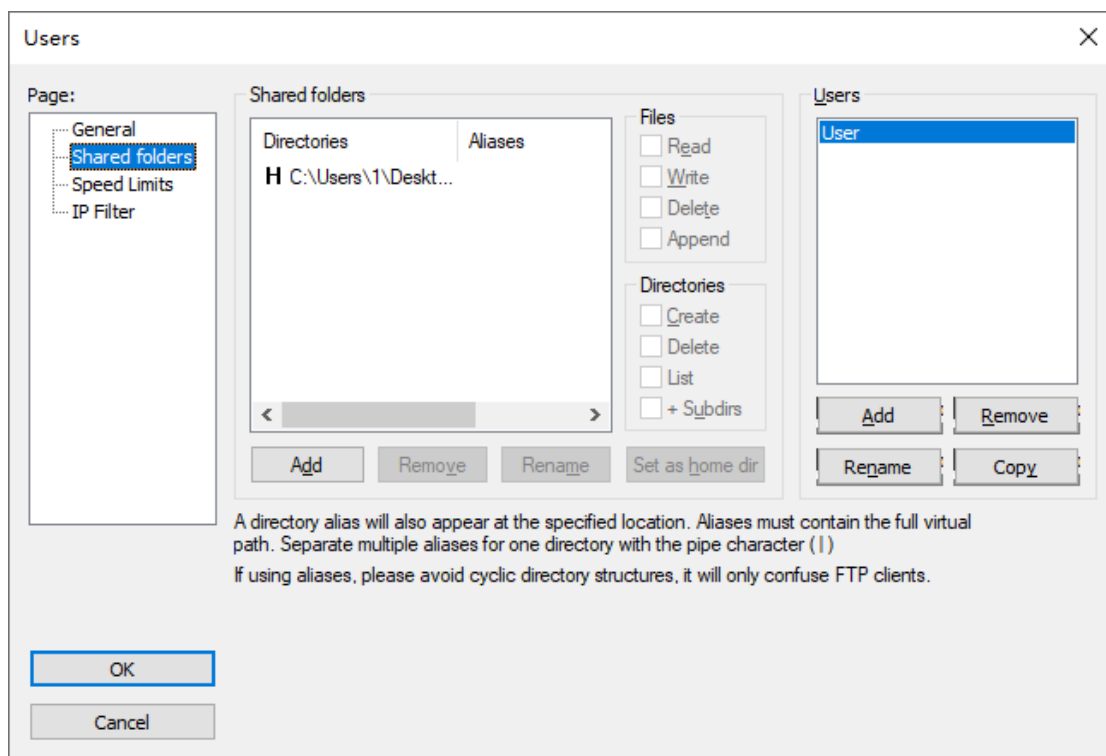


图 4-2 设置服务器目录

配置完毕后，在 WINDOWS 资源管理器中输入 ftp://本机的 IP 地址，如果可以连接 FTP 服务器，就表示配置成功。

4.2 远程升级

调用 LUA 脚本 API 中的 start_upgrade() 函数下载 PKG 包，示例工程中画面配置及 LUA 程序如下：

```
137 -- start_upgrade("此处填写FTP工程文件地址")
```

串口屏会在下载完成后自动更新并重启。

5. 如何使用 TCP 协议进行 SOCKET 通信

本章节主要介绍以下 4 点：

- (1) NetAssist 配置；
- (2) 工程配置；
- (3) SCOKET 屏幕收；
- (4) SCOKET 屏幕发。

本例程中，使用 NetAssist 网络调试助手调试工具和屏幕通讯。在本例程压缩包已包含，

也可以自行网上下载。如图 5-1 所示。

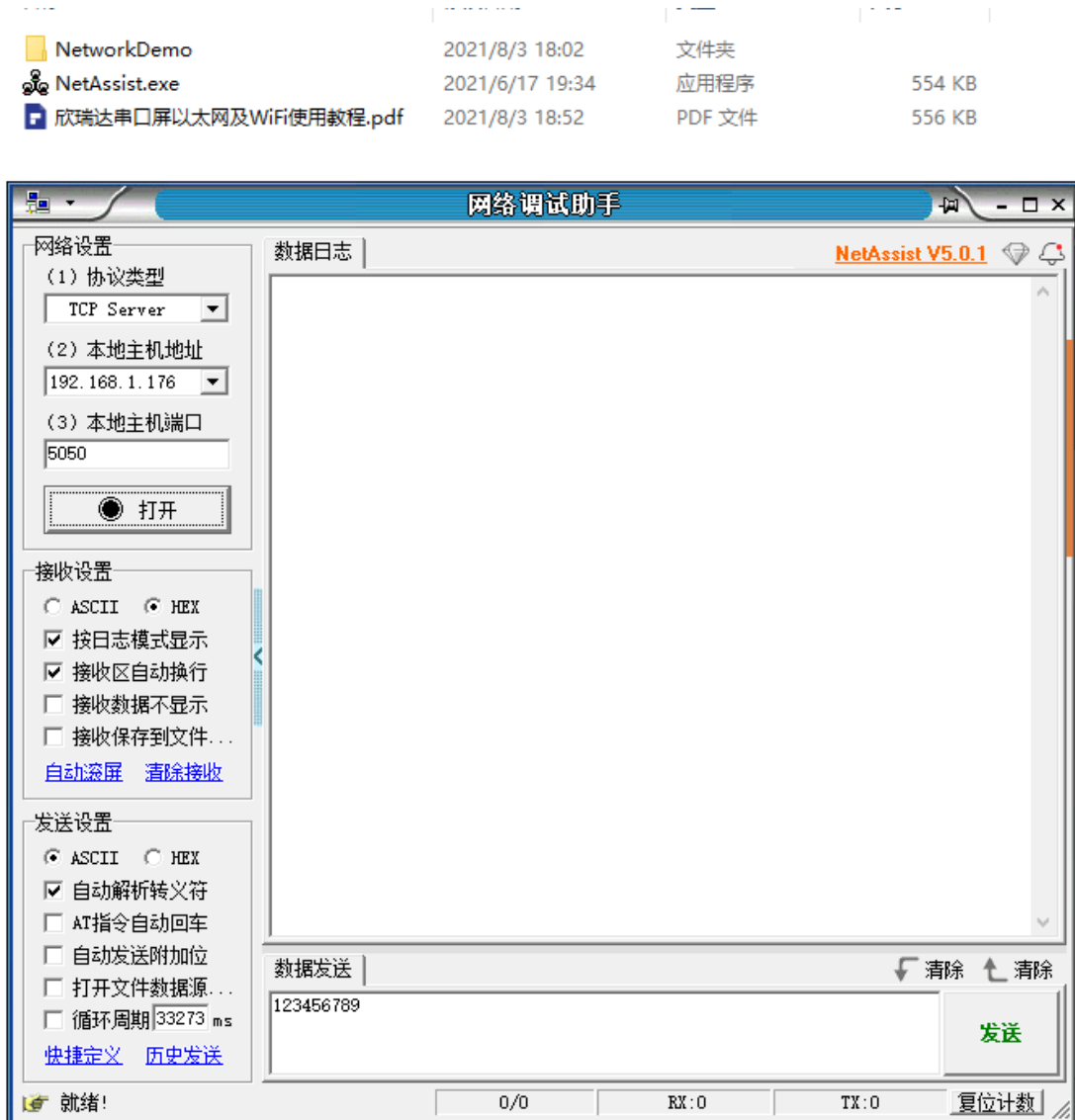


图 5-1 网络调试助手

5.1 NetAssist 配置

本例程中使用 NetAssist 网络调试助手演示，配置如下：

- 1) 协议类型：TCP Server；
- 2) 本地 IP 地址：根据用户网络为准；
- 3) 本地端口号：本例程中默认是“5050”；
- 4) 连接对象：根据用户网络为准；
- 5) 打开：打开连接，如图 5-2 所示。



图 5-2 NetAssist 配置

注意：NetAssist 网络调试助手详细使用教程可在网上查找资料了解。

5.2 工程配置

在工程属性中，配置网络以下属性：

- 1) 网络：网卡模式；
- 2) SSID 名称：Test(以太网接口可不填)；
- 3) 连接密码：11223344(以太网接口可不填)；
- 4) DHCP：启用；
- 5) 网络服务：客户端；
- 6) 服务端口：5050。

属性配置如图 5-3 所示。

网络	
网络	网卡模式
SSID名称	Test
密码	11223344
DHCP	允许
DNS服务器	192.168.1.1
网络服务	客户端
远程服务器	192.168.1.176
服务端口	5050
透传	否

图 5-3 网络配置

5.3 修改服务器 IP、PORT

1. 画面配置

在画面中，添加两个文本控件（控件 ID1、控件 ID2，系统键盘输入）和一个按钮控件（控件 ID5），其中控件 ID1 显示“服务器 IP”、控件 ID2 显示“端口”。控件 ID5 为按钮按下时获取当前屏幕配置的“服务器 IP”和“端口”。如图 5-4 所示。

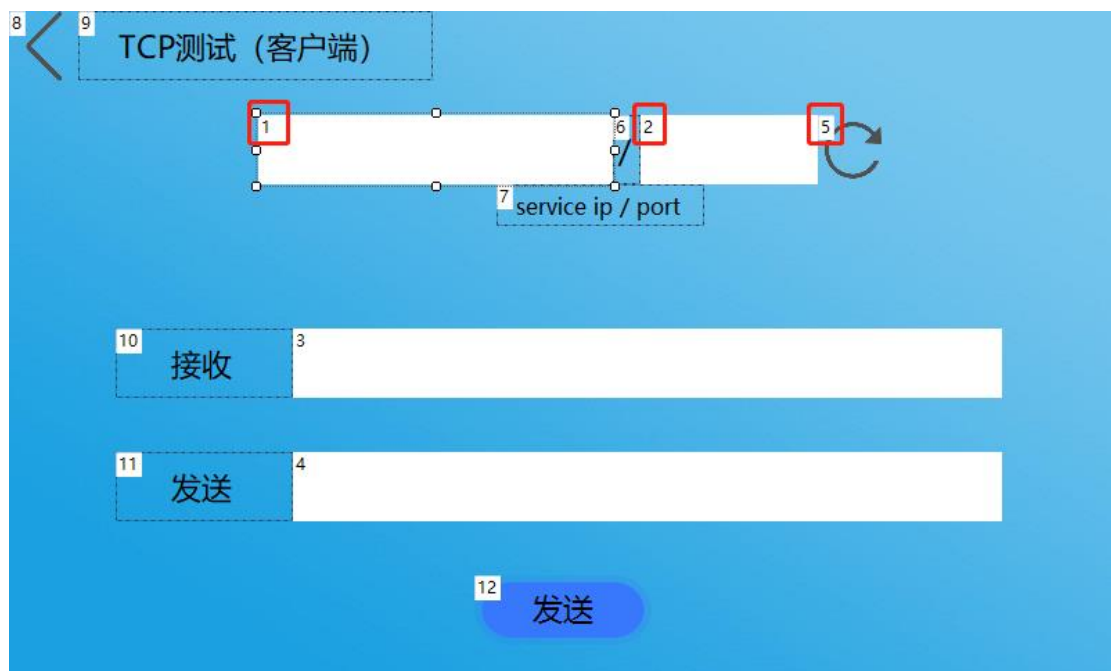


图 5-4 画面配置

2. LUA 脚本编辑

本例程中，获取和设置 port、server IP 的代码如下：

```

116 elseif screen == sc_TCPTest then
117     if control == 1 or control == 2 then --设置网络服务参数
118         local service_ip = get_text(screen, 1)
119         local service_port = get_text(screen, 2)
120         set_network_service_cfg(0, 1, service_port, service_ip) --0默认模式, 1客户端模式
121         save_network_cfg()
122
123     elseif control == 5 and value == 0 then --获取网络服务参数
124         local wificom, mode, port, server_addr = get_network_service_cfg()
125         set_text(sc_TCPTest, 1, server_addr)
126         set_text(sc_TCPTest, 2, port)

```

核心 API 函数

1) set_network_service_cfg(wificom, mode, port, server_addr)

设置网络服务参数

wificom -默认为 0, 为 1 时启用透传模式（即无线串口屏）

mode -0 禁用网络服务, 1 客户端模式, 2 服务器模式

port -服务端口

server_addr -服务器地址(屏作客户端时)

2) get_network_service_cfg()

获取网络服务参数, 返回 4 个参数, 说明同上

wificom, mode, port, server_addr = get_network_service_cfg()

基本思路: 当按钮控件 ID5 松开的时候, 触发触摸控件回调函数 on_control_notify(), 再调用 get_network_service_cfg () 获取当前屏幕的网络服务配置, 并显示在 ID1、ID2 的文本控件中。当对文本控件 ID1、ID2 输入的时候, 调用 set_network_service_cfg() 修改配置和保存 save_network_cfg() 即可。

5.4 SOCKET 屏幕收

1. 画面配置

在画面中, 添加 1 个文本控件 (控件 ID3), 作为显示接收到的服务器数据, 配置如图 5-5 所示。

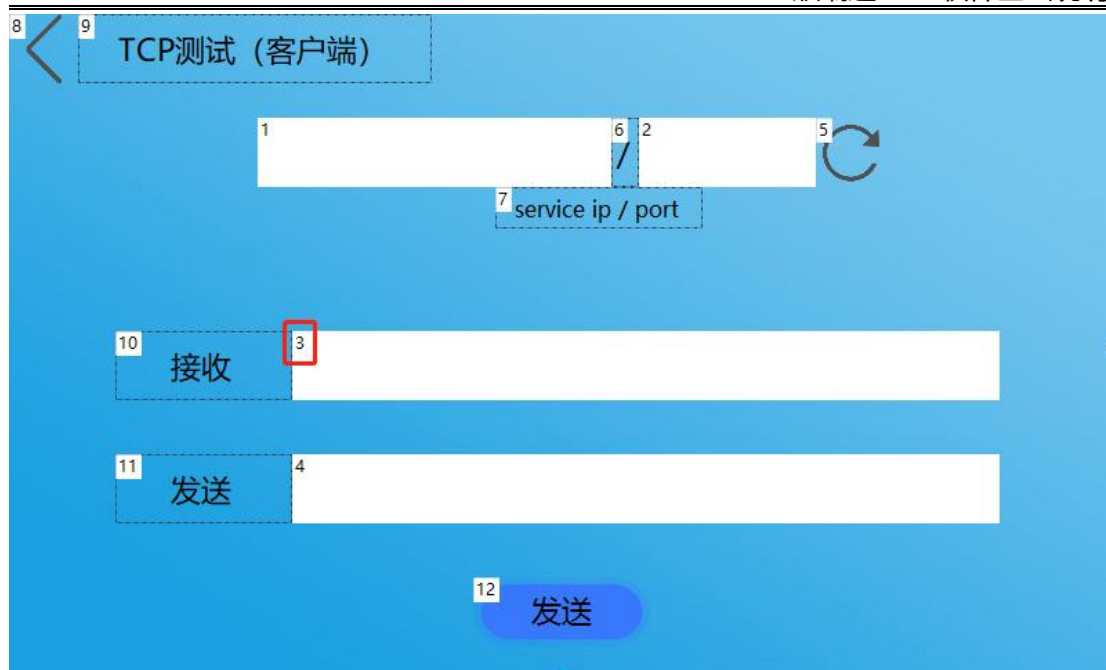


图 5-5 画面配置

2. LUA 脚本编辑

本例程中，将屏幕接收到服务器的数据以 16 进制的格式显示出来，代码如下：

```

212 -- 当客户端 SOCKET 接收到数据时，系统自动回调此函数。
213 function on_client_recv_data(packet)
214     local recv_buf = string.format("%02X",packet[0])
215     for i= 1, #(packet) do
216         recv_buf = recv_buf .. " " .. string.format("%02X",packet[i])
217     end
218
219     set_text(sc_TCPTest, 3, recv_buf)
220
221     -- 返回1时，数据不通过串口发送给用户MCU
222     return 1
223 end

```

核心 API 函数

1) on_client_recv_data(packet)

当客户端 SOCKET 接收到数据时，系统自动回调此函数。

packet: 接收的字节数组 packet，下标从 0 开始

基本思路：当服务器向屏幕发送数据时，屏幕自动回调 on_client_recv_data()。本例程中，对接收服务器的数据仅作显示。实际应用中，用户可根据服务器的数据响应不同的动作，如更新文本、设置按钮、进度条、通过串口给 MCU 动作等等。

5.5 SOCKET 屏幕发

1. 画面配置

在画面中，添加 1 个文本控件(控件 ID4, 系统键盘输入)和 1 个按钮控件(控件 ID12)，

当用户点击文本控件弹出键盘输入数据，点击“发送”按钮将文本控件的数据发送至服务器，画面配置如图 5-6 所示。

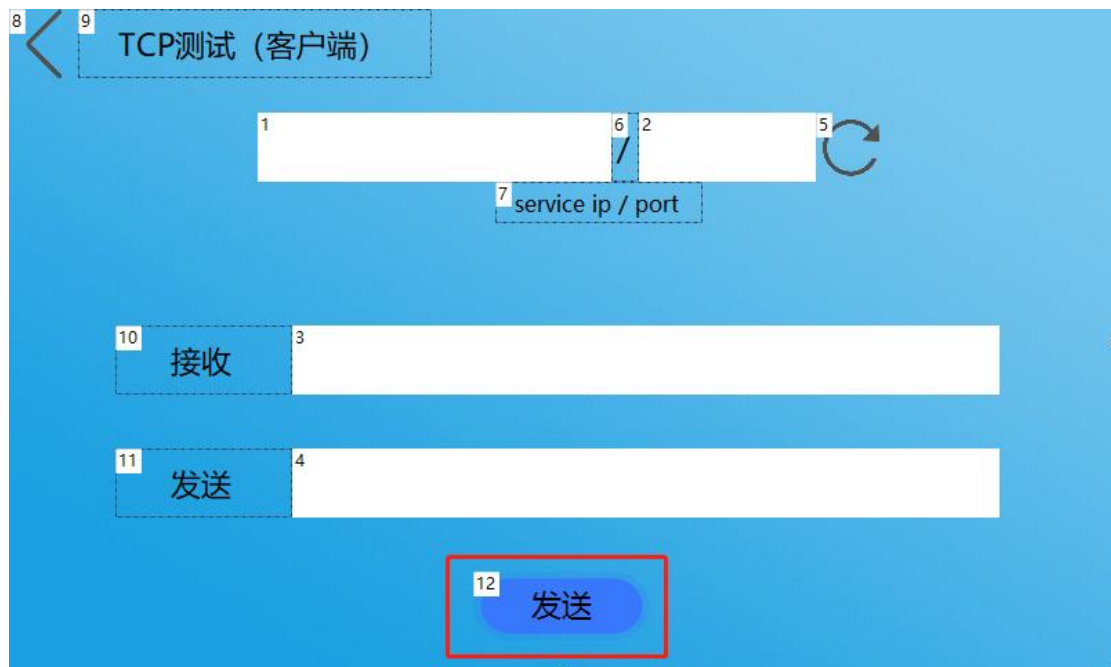


图 5-6 画面配置

2. LUA 脚本编辑

本例程中，在获取文本控件文本后，以 16 进制发送到服务器，代码如下：

```

200  -- 向服务器发送字符串
201  function my_send_to_server(send_data)
202      local send_buf = {}
203      local send_len = string.len(send_data)
204      for i = 1, send_len
205      do
206          send_buf[i - 1] = tonumber(string.byte(send_data, i, i))
207      end
208
209      client_send_data(send_buf)
210  end

128  elseif control == 12 and value == 0 then --向服务器发送数据
129      my_send_to_server(get_text(screen, 4))
    
```

核心 API 函数

1) client_send_data(packet)

通过客户端 SOCKET 发送报文到服务器

packet: 发送字节数组 packet，下标从 0 开始。