

DSP 本地/远程网络通讯软件说明

一. 功能说明

本方案采用 DSP 的 SPI 接口连接 CH395 芯片，实现 DSP 的 TCP/IP 通讯，实现了 DSP 的本地/远程低速数据上报、本地实时高速数据上报、实时/历史波形显示与存储、DSP 本地/远程控制、DSP 本地/远程代码升级等功能。

在本地/远程低速数据请求时，数据上报速率与网络情况相关，通常而言 20-100ms 可请求并返回一次数据，可根据内存地址进行寻址，一次请求返回 2 个 Byte。本地高速数据上报功能，在开关频率为 10KHz 且 SPI 在 10Mbps 速率下最高能够达到 568.18KByte/s 的速率上报，即每个开关周期最高可上报 58 个 Byte，SPI 的速率提升仍有很大空间，因此数据发送量的提升空间仍很大。高速数据测速如下图 1.1 所示。

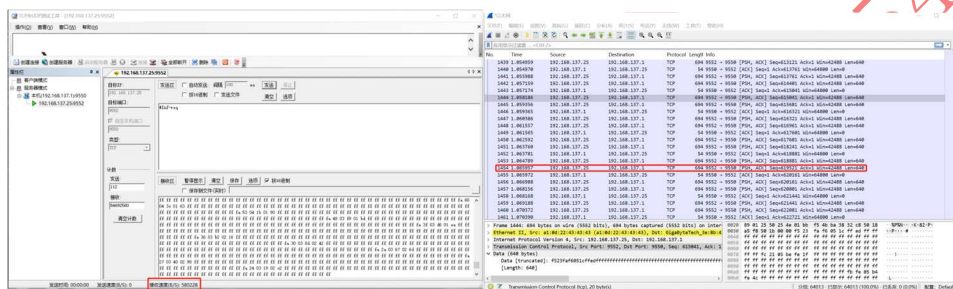


图 1.1 高速上报速率测试及抓包分析

上位机对于数据的实时/历史波形显示时间戳精度在本地高速数据请求下可达 us 级，如图 1.2 所示，在本地/远程低速数据请求可达 ms 级。

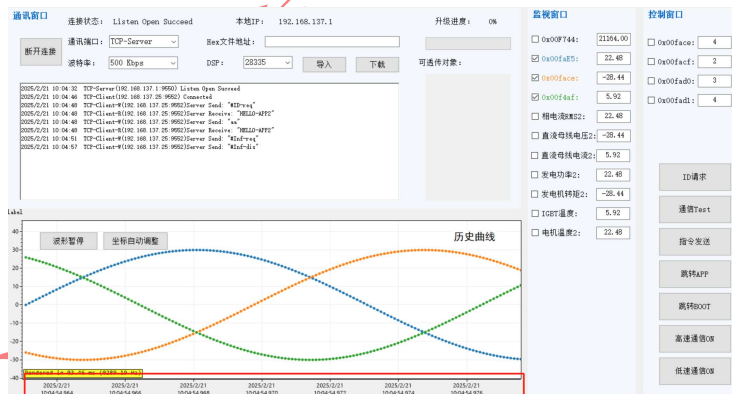


图 1.2 实时/历史波形图

历史曲线皆保存为.csv 文件，文件保存在上位机根目录下，.csv 文件可由 Excel/Matlab 直接打开并进行数值分析，如图 1.3 所示。

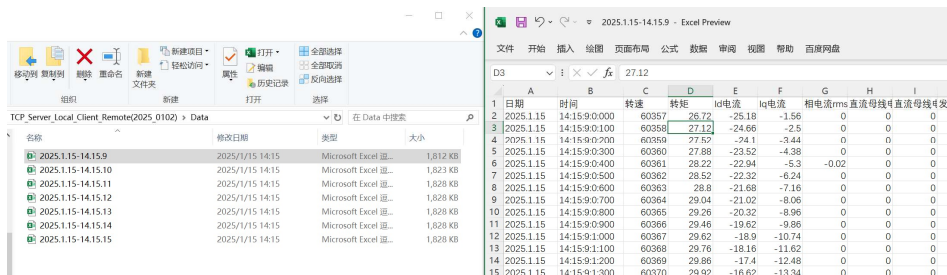


图 1.3 历史数据存储

上位机可发送 DSP 内存数据更改请求，可根据任意输入地址寻址，请求更改 DSP 指定地址内存数据，如下图 1.4 所示。

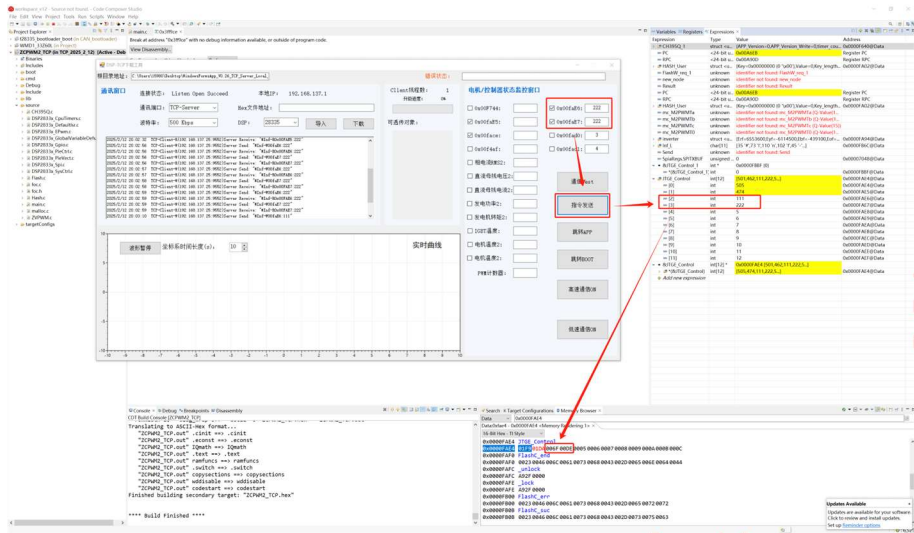


图 1.4 内存数据更改

上位机可实现本地/远程程序更新，通过选择 CCS 工程编译获得的 HEX 文件进行代码的更新，下载流程完整可靠，具有校验，永不 BAN 机。

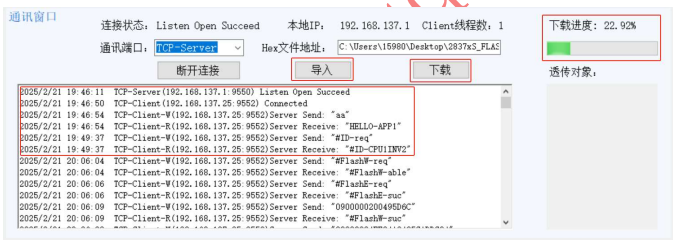


图 1.5 程序下载

注：远程功能需租用 TCP 私有服务器。

二. 硬件准备

本方案采用 DSP 的 SPI 接口连接 CH395 芯片，实现 DSP 的 TCP/IP 通讯。

用户可以有两种连接方式：1.在设计阶段，就将 CH395 布置在 PCB 中,且与 DSP 连接；
2.在设计阶段，预留一个 DSP 的 SPI 接口至设备接口处(不可过长 10cm 以内)，使用时将 CH395 模块连接预留的接口。

方式 1 需要在 PCB 板内布置 CH395 芯片，固件版本使用 09 及以上，Q/L 两种封装都可用。原理图可参考下图 2.1 所示，或可直接参考沁恒官网原理图。

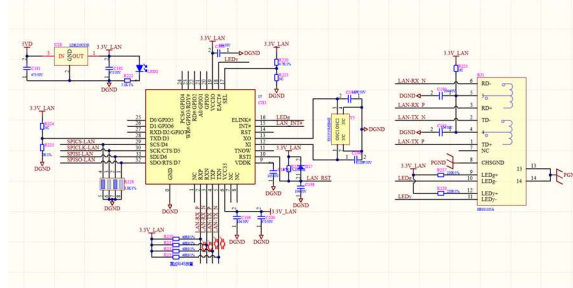


图 2.1 CH395 原理图

方案 2 在预留 SPI 接口后，采购 CH395 模块并在使用中连接模块即可，模块如下图 2.2 所示。

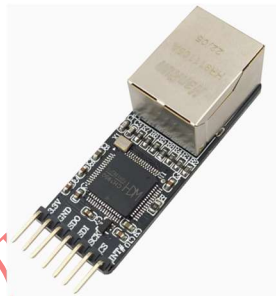


图 2.2 CH395 模块

这两种方式下，DSP 都只需一个标准 SPI 接口连接 CH395 的 SDO、SDI、SCK、CS 即可，并且该 SPI 接口不可连接其他芯片。

注：1.更多芯片信息请查看官网：[CH395DS1.PDF](#) - 南京沁恒微电子股份有限公司

2.模块采购链接：[CH395Q 模块 UART 串口透传/SPI 接口 MCU 100M 以太网 TCP/IP 协议-淘宝网](#)

三. TCP 连接方式说明

由于网口连接的灵活性，本方案可有多种 DSP 的连接方式，例如有线、无线、远程，且皆可进行 DSP 的下载和监控。首先是本地连接方式，如下图 3.1 所示，图(a)为网线有线连接方式，CH395 网口直接连接 PC；图(b)为 CH395 网口连接路由器，PC 通过有线或无线方式连接同一路由器(无线连接时，PC 连接路由器 WIFI 即可)。

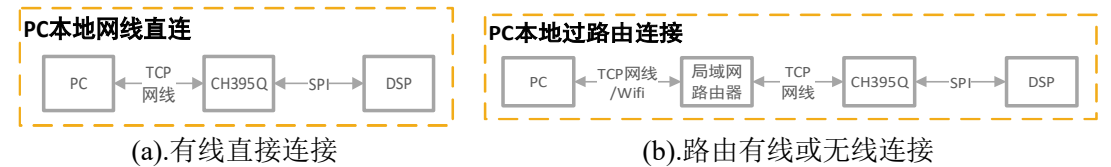


图 3.1 DSP 本地连接方式

同样远程连接方式也有很多种如图 3.2 所示。个人电脑通过 WIFI、热点、路由器等方式连接至互联网，CH395 网口通过路由器或 PC 共享网络连接至互联网。

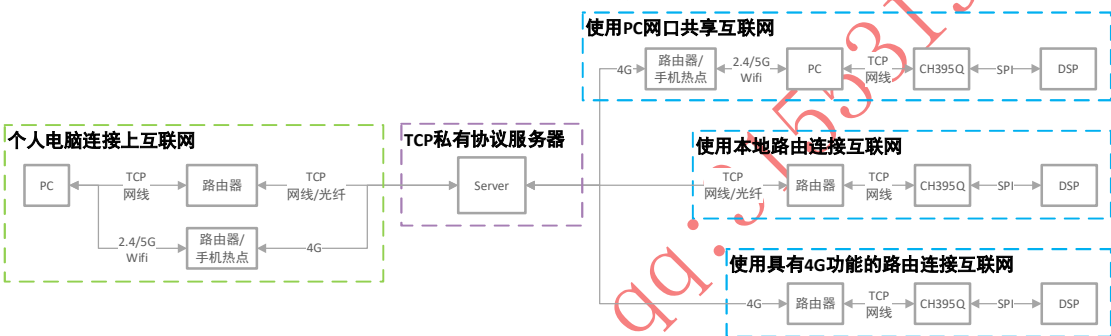


图 3.2 DSP 远程连接方式

注： TCP 私有服务器需租用并搭载服务器应用。

四. 工程地址分配

如下图 4.1 所示，该工程通过两个宏定义来控制编译，从而控制程序运行模式、编译地址以及编译 APP。

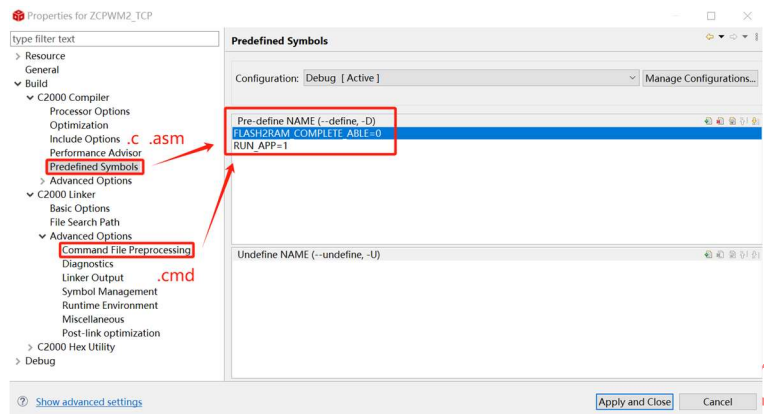


图 4.1 工程编译配置

工程分为两种运行模式：1.配置 FLASH2RAM_COMPLETE_ABLE=1，则代码全部 copy 至 RAM 中运行；2.配置 FLASH2RAM_COMPLETE_ABLE=0，则仅将 CODE_SECTION 至 ramfuncs 的代码 copy 至 RAM 中运行，其余代码在 Flash 运行，例如下图 4.2 所示；

```
/*
*****
RAM ALLOCATION (N/A)
*****
*/
#if FLASH2RAM_COMPLETE_ABLE == 0
#pragma CODE_SECTION(hash_insert, "ramfuncs")
#pragma CODE_SECTION(hash_find, "ramfuncs")
#endif
```

图 4.2 运行编译宏

工程地址分为两个 APP，其中 APP1 作为 bootloader，配合上位机将代码烧录至 APP2 地址区域：配置 RUN_APP=1 则编译至 APP1 地址区域；配置 RUN_APP=0 则编译至 APP2 地址区域，如下图 4.3 所示

```
#if RUN_APP == 1
SECTIONS
{
    FLASH2RAM_PART
    APP1
}
```

图 4.3 cmd 编译宏

其具体地址区域分配在 DSP 工程文件“F28335_FLASH2RAM_COMPLETE.cmd”或“F28377S_FLASH2RAM_COMPLETE.cmd”的中。

五. Boot 结构

DSP 在上电或复位后代码首先进入 ROM 固件,执行一次 boot“ROM_boot”,其通过 GPIO 选择为 Flashboot 模式;

然后一次 boot 引导程序指针指向固定地址“codestart”, 从后进入二次 boot。“codestart”其代码在“xxxxx_CodeStartBranch.asm”中,其地址在 28335 为 0x33FFF6,28377S 为 0x080000, 由 cmd 控制。“codestart”实际上为一条跳转指令, 占用内存 16bits(DSP 最小地址单位), 该指令跳转至“xxxxx_CodeStartBranch.asm”中的“wddisable”。

进入代码“xxxxx_CodeStartBranch.asm”中的“wddisable”, 代码首先关闭 Watch dog, 377S 还关闭了 ECC, 然后跳转至代码“xxxxx_SectionCopy_nonBIOS.asm”。

在“xxxxx_SectionCopy_nonBIOS.asm”中首先进行 APP 跳转判断。APP 的跳转条件为地址 0x337FFF/0x0BFFFF(FLASH 用户数据扇区 B/N 的最后 16bit)中值是否为 0xffff, 若是则跳转 APP1 的 copy, 若不是则跳转 APP2 的 codestart, 同样在 APP2 的“xxxxx_SectionCopy_nonBIOS.asm”也会进行一次 APP 的跳转条件判断。跳转结束后接着进行 Flash 至 RAM 的代码/数据搬运工作,此时根据宏定义“FLASH2RAM_COMPLETE_ABLE”分为全代码 copy 和部分代码 copy, copy 完成跳转至代码“boot28.asm”。

“boot28.asm”代码为堆栈指针初始化、系统状态寄存器初始化、c 语言环境初始化, 完成后跳转至“main.c”函数执行 APP 代码。

下图 5.1 为代码 boot 引导流程图。在实际使用中, 可直接代码跳转 APPx 或刷写 0x0337FFF/0x080000 内容后重新上电/复位 DSP 自动跳转。

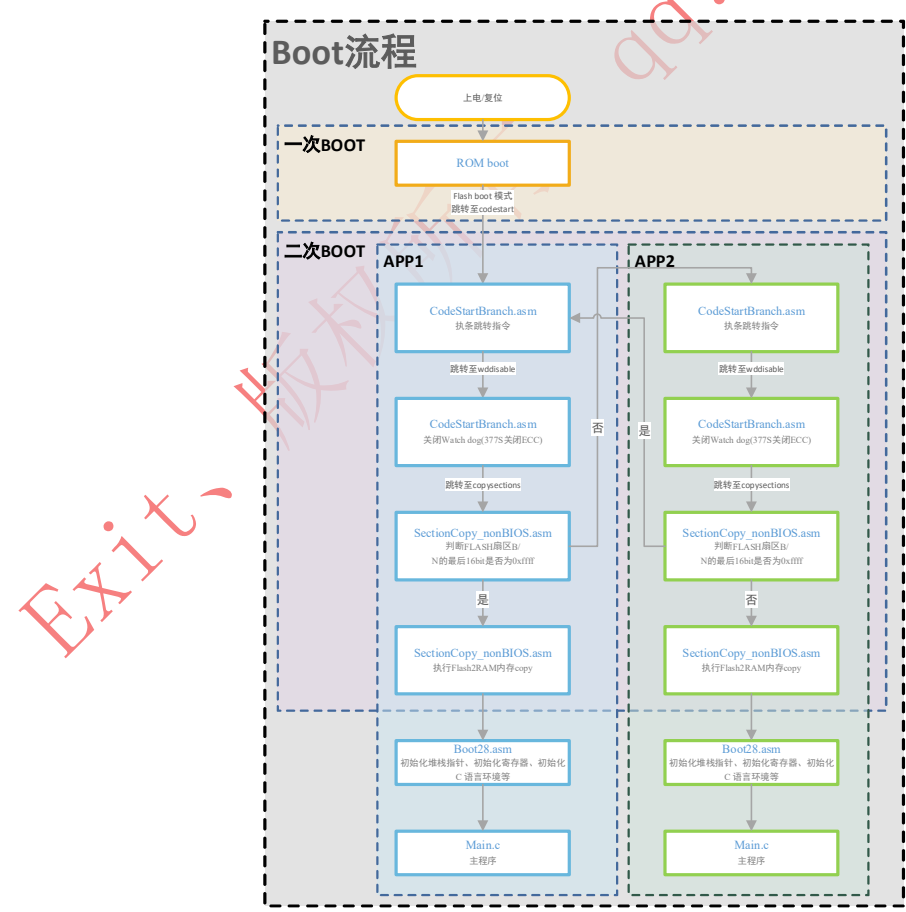


图 5.1 BOOT 流程图

六. TCP 通讯协议

本地模式下个人电脑(上位机)作为 TCP 服务器进行端口监听，DSP 作为 TCP 客户端主动连接上位机，上位机/DSP 都可主动断开连接，目前上位机 IP 固定为 192.168.137.1:9550，DSP IP 固定为 192.168.137.25:9552。

使用远程服务器时，上位机需连接互联网且启动获取 IP，DSP 也可设置自动获取 IP，此时上位机与 DSP 都作为 TCP 客户端主动连接服务器。

除高速信息外，协议主要采用一问一答方式通讯，字符采用 ASCII 编码，同时通讯指令皆以“#”字符开头。若在一定时间内，DSP 未回应上位机指令，则上位机主动断开 TCP 连接。TCP 协议主要有三个部分的指令，分别为 APP 升级、数据监控、连接测试，如下表所示：

功能	上位机发送	DSP 回复	备注
APP 升级			Flash 烧录请求
			Flash 擦除请求
			Flash 烧录
			刷写结束，进入校验
			Flash 校验
			校验结束
			跳转 APP2 指令
			跳转 APP1 指令
			TCP 的 CPU 权限交换
数据监控			数据更改
			低速数据请求
			高速数据请求
			高速数据结束
连接测试			DSP ID 请求
			DSP 信息请求

- 1、在连接测试中，
 - 2、“#Inf-req”、“#Inf-dis”分别为
 - 3、在 APP 升级中，上位机首先
- TCP 通讯流程图如下图 6.1 所示：

图 6.1 TCP 通讯流程图

注：TCP 协议除上述内容外可增加 Modbus-TCP 等协议内容。

七. DSP 程序架构

DSP 程序架构如下图 7.1 所示，这里使用了中断嵌套。主要由于 SPI 的 FIFO 长度有限，为了提高 SPI 在高速通讯时的总线满载率而设计的。

DSP 程序嵌套时序如下图 7.2 所示，从时序上来看就是在 FIFO 发送完成后进入 RX 中断立刻再次给 FIFO 装载数据接着发送。而由于 TCP 发送函数嵌套在 EPWM 中断的头部，因此合理设计发送长度，SPI_RX 中断不会发生在 main 中，这于 SPI 原理相关，SPI 全双工发送多少 Bytes 必然接收多少 Bytes。

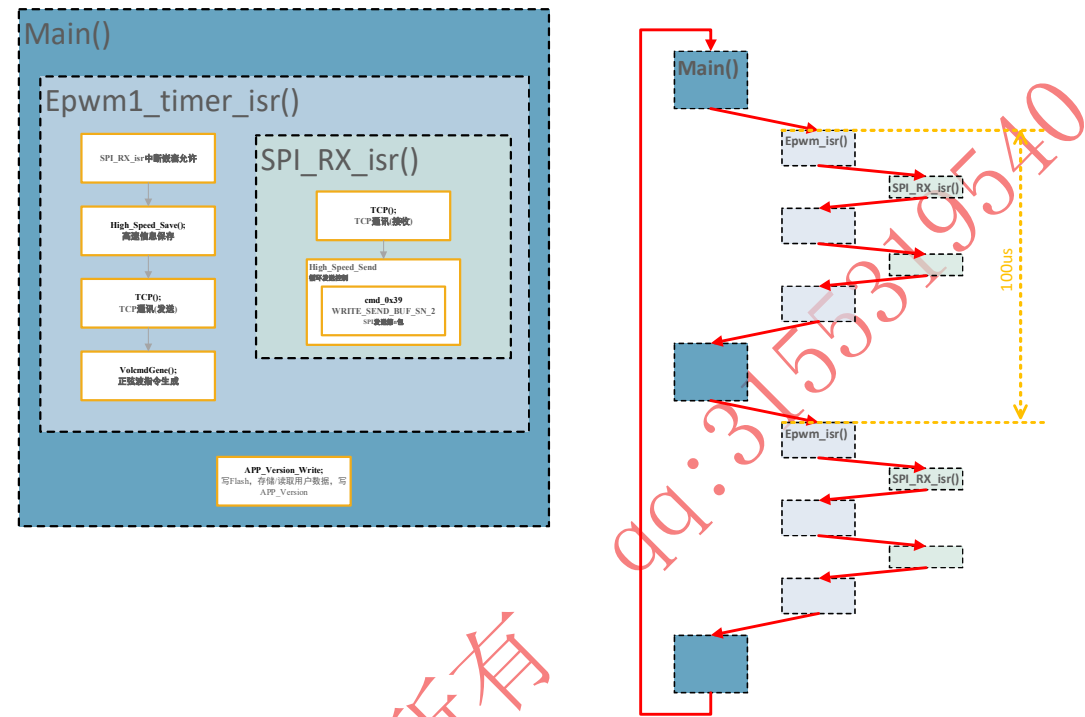


图 7.1 程序架构

图 7.2 程序嵌套时序

八. 低速通讯流程

如下图 8.1 所示为 TCP 通讯流程图，流程图较为完整的绘制了 DSP 通过 SPI 端口连接 CH395 时进行低速通讯的控制流程，下载 APP 升级、低速数据监控、连接测试皆属于低速通讯，该流程无法表征高速通讯时的流程。

图 8.1 TCP 低速通讯流程图

Exit、版权所有 qq:3155319540

九. 高速通讯流程

在高速信息发送中, 采用了双缓异步存储、强制同步发送的方式, 进行存储和发送, DSP 开了两片数据缓存区 High_Speed_Inf_1 和 High_Speed_Inf_2。首先将 11 个开关周期的数据完整存储在 High_Speed_Inf_1 中, 存储完成后切换存储对象为 High_Speed_Inf_2, 紧接着将在 8 个开关周期内发送完毕 High_Speed_Inf_1 中的数据, 余 3 个开关周期判断是否有 TCP 接收。也就是说此时 11 个开关周期已过, High_Speed_Inf_2 已存满, 开始发送 High_Speed_Inf_2, 以此循环, 这样做的目的是为了使发送存储互不干扰且可分别设计发送和存储时序, 达到异步操作同步处理的效果, 下图 9.1 所示为高速时序流程图。

图 9.1 TCP 高速通讯流程图

下图 9.2 为开关周期 100us 时, High_Speed_Inf 的存储结

图 9.2 10Mbps 下全带宽存储结构图

下图 9.3 为 High_Speed_Inf 的发送时序, 可见数据发送占用 8 个

图 9.3 10Mbps 下全带宽发送时序图

SPI 带宽全跑满的话, 虽然数据量大、速率高, 但由
阉割后的 High_Speed_Inf 的存储结构如下

图 9.4 10Mbps 下阉割后存储结构图

下图为阉割后 High_Speed_Inf 的发

图 9.5 10Mbps 下阉割后发送时序图

十. 上位机说明

在开启上位机 TCP 连接前，首先选择端口模式，本地模式下选择 TCP-Server，远程模式下选择 TCP-Client，端口选择下拉框如下图 9.1 所示。

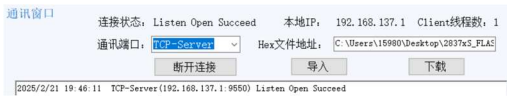


图 9.1 通讯端口选择

在点击连接按钮前，若使用本地模式则将需要先更改本地网卡的属性，将网卡 IP 配置与上位机配置文件“Properties.ini”(在上位机根目录下)中“String_Tcp_Server_IP”配置的 IP 一致(12.168.137.1)，如下图 9.2 所示设置 PC 网络连接。同时需要注意确保 DSP 代码与“Properties.ini”中的服务器 IP 和端口号保持一致。配置完成后点击连接按钮，若连接状态与信息窗口皆提示“Listen Open Succed”则说明上位机作为本地 TCP server 打开成功。

如果使用远程模式，则需要在上位机配置文件“Properties.ini”中修改服务器 IP 和端口号(租用的服务器地址)，客户端的可以不管，然后更改本地网卡的属性，将网卡 IP 设置为自动获取 IP 地址，使 PC 连接上互联网即可。同时注意确保 DSP 中配置的服务器地址要与配置文件“Properties.ini”中的服务器 IP 和端口号保持一致。

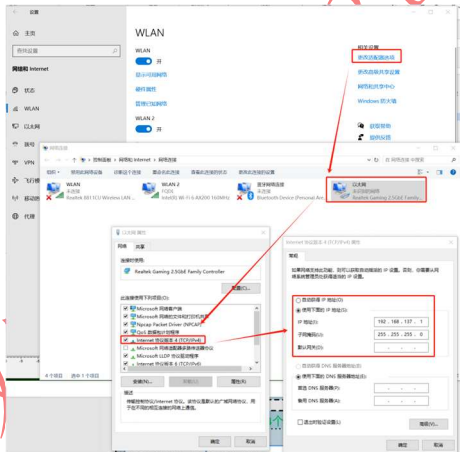


图 9.2 PC 网络设置

上位机在接收到低速/高速数据时会进行数据的解析和存储，数据皆存储在上位机根目录的 Data 文件夹中，按照数据时间存储为.csv 文件，文件中都有数据的时间戳。

上位机的实时/历史曲线皆从 Data 文件夹按照时间进行检索，为了防止过多数据加载导致软件卡顿，因此高速模式下实时曲线刷新频率为 1HZ，低速模式可配置曲线刷新时间，更改配置文件“Properties.ini”中的 Simple_Time 即可，单位为 ms。

实时/历史曲线的切换通过“波形暂停”按钮控制，曲线可用鼠标右键拖拉缩放，鼠标左键拖拉坐标，以及鼠标滚轮进行缩放。

右侧监窗口可以选择通过点选来控制曲线窗口的显示对象，且可右击其名称可进行重命名，低速数据仅会按照“0x00xxxx:”的格式进行解析并进行下发（例如“0x003D0E:”，不区分中文或英文冒号），若使用不可解析格式命名的话，低速模式即使点选了也不会下发和采集，高速数据与该地址无关，高速数据的存储指针由 DSP 代码赋予。此处的点选名称在更改且进行数据采集时，其名称都会在.csv 的存储中更改。监视窗口与曲线窗口如下图 9.3 所示。

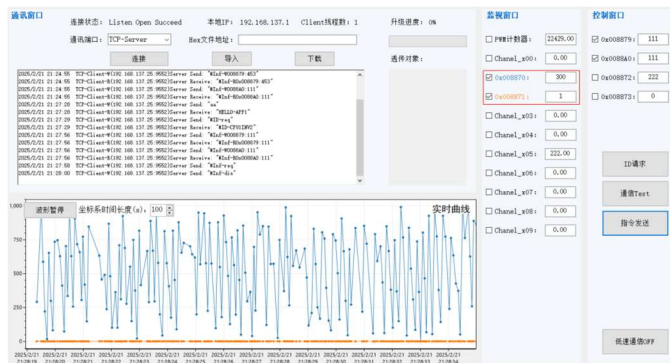


图 9.3 监视窗口

右侧控制窗口，可通过配置下发地址以及数据，点击“指令发送”按钮后向 DSP 发送数据更改请求，DSP 更改数据后回复（注意不要写 Flash 地址、ROM 地址或一些不可更改的系统寄存器地址）。同样右击其名称可进行重命名，且仅会按照“0x00xxxx:”的格式进行解析并进行下发，若使用不可解析格式命名的话，同样即使点选了也不会下发。控制窗口及信息窗口返回结果如下图 9.4 所示。



图 9.4 控制窗口

十一. 下载操作说明

下面对上位机下载的操作流程进行解释，下载窗口如下图 9.4 所示：

- 1) 首先在上位机点击“连接”按钮且提示“Listen Open succeed”后为 DSP 上电，DSP 上电自动搜索上位机并建立连接后，信息窗口提示“Connected”，此时 DSP 已连接至上位机；
- 2) 接着点击“ID 请求”按钮，根据信息窗口的返回信息确认连接对象是否为目标 DSP，且测试 DSP 连接稳定性。若发送断开连接，信息窗口提示“Connected Failed”，此时上位机无需操作，复位或重新上电 DSP 即可再次尝试创建连接（上位机首次打开端口可能会出现连接失败，不要关闭上位机，给 DSP 重新上电再次尝试即可）；
- 3) 然后点击“通讯 test”按钮，根据信息窗口的返回信息确认连接对象处于 APP1 中，若不在则需要点击“跳转 Boot”按钮后重新上电连接，并再次判断是否处于 APP1 中。
- 4) CCS 编译工程后自动生成 Hex 文件在，Hex 文件在工程目录/Debug 中，文件名称一般为“工程名称+.hex”，点击“导入”按钮并选择想下载的 Hex 文件，选择后文件路径会显示在“Hex 文件地址”中。
- 5) 点击“下载”按钮，自动进行下载流程，通过配置“Properties.ini”中的“TCP_FlashW_Information_Visible”、“TCP_FlashC_Information_Visible”可设置下载过程中的报文是否显示在信息窗口中，开启信息显示后下载速度会变慢，下载具有百分比和进度条显示。
- 6) 下载成功后会自动跳转 APP2，从信息窗口可见 DSP 重新进行了连接，即自动进入 APP2。若下载失败，可重新上电重新尝试下载，通常下载失败的原因是 Flash 目标地址非 APP2 地址导致的(cmd 配置错误)，可开启报文信息显示进行判断分析。

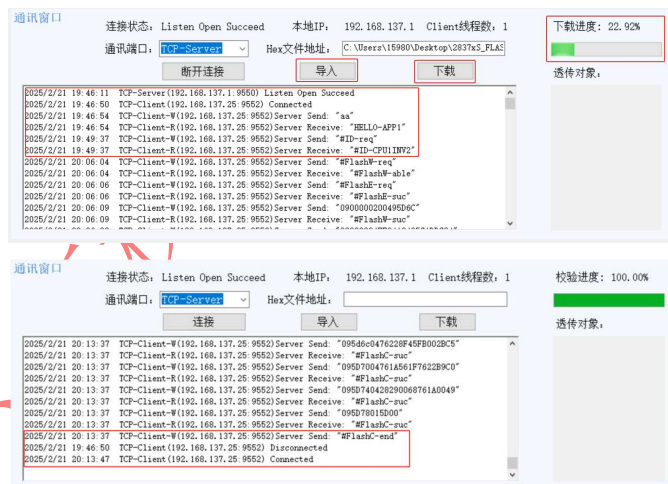


图 9.4 下载窗口

十二. 移植说明

建议往提供的框架内移植功能，而不是将功能移植到现有逆变器工程代码。

向现有逆变器工程代码移植 TCP 功能步骤：

注：要注意 DSP 的低速时钟速率，并将 SPI 速率配置为 10Mbps

Exit、版权所有 qq:3155319540