Kinco 步科

自由口网口通讯

2021

Kinco步科

1 自由口网口的定义以及应用场景

2 步科自由口使用说明

3 使用注意事项

Kinco 步科

一、自由口网口的定义以及应用场景

自由口通讯的定义

网口通信是通过网络将各个孤立的设备进行连接，通过信息交换实现人与人，人与计算机，计算机与计算机之间的通信。自由口网口通讯主要应对两台及以上设备没有使用行业内标准的通讯协议，导致无法通讯的情况，自由口通讯需要自己定义协议内容，所以自由口通讯也叫无协议通讯。

当使用网口进行自由口通讯时，需要接口为RJ45，设置触摸屏IP地址与设备IP处在同一网段，获取设备的端口号、网口协议内容以及报文格式等。

自由口通讯的应用场景

1. 扫码枪以及RFID系统、数据传输
2. 没有使用行业标准的通讯协议的仪表，外围设备间通讯
3. 需要对数据进行一定量的解析，自定数据格式的场合

Kinco 步科

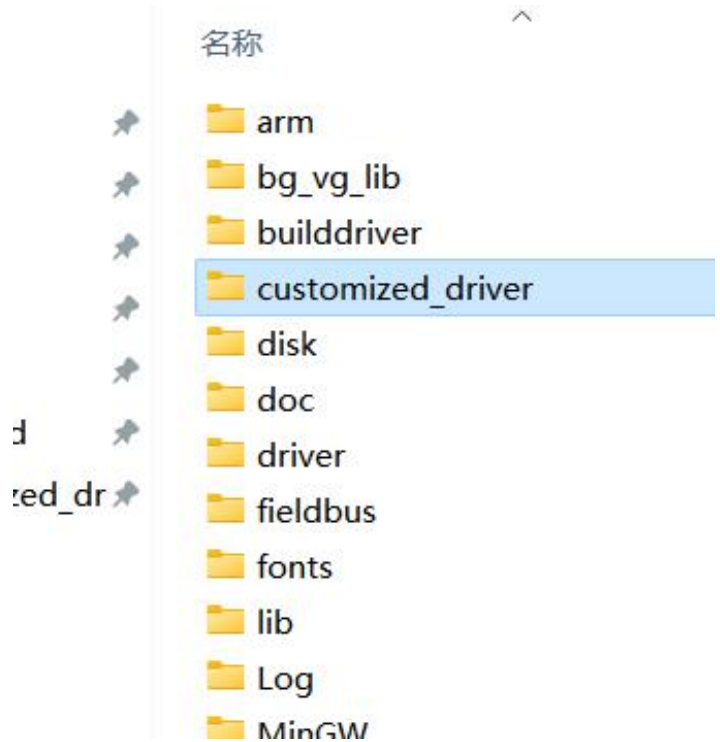
二、使用说明

步科自由口使用说明

网口自定义协议的驱动目前可以支持一个（双网口HMI支持任意一个网口做自由通讯）定制自由协议，通讯协议选择Customized Ethernet protocol。选择触摸屏后，进入网络通讯配置，设置触摸屏IP地址、端口号即可，协议通过LW和LB寄存器进行数据的交换。若是第一次使用驱动，请先按照后续介绍完成驱动安装。

驱动使用步骤

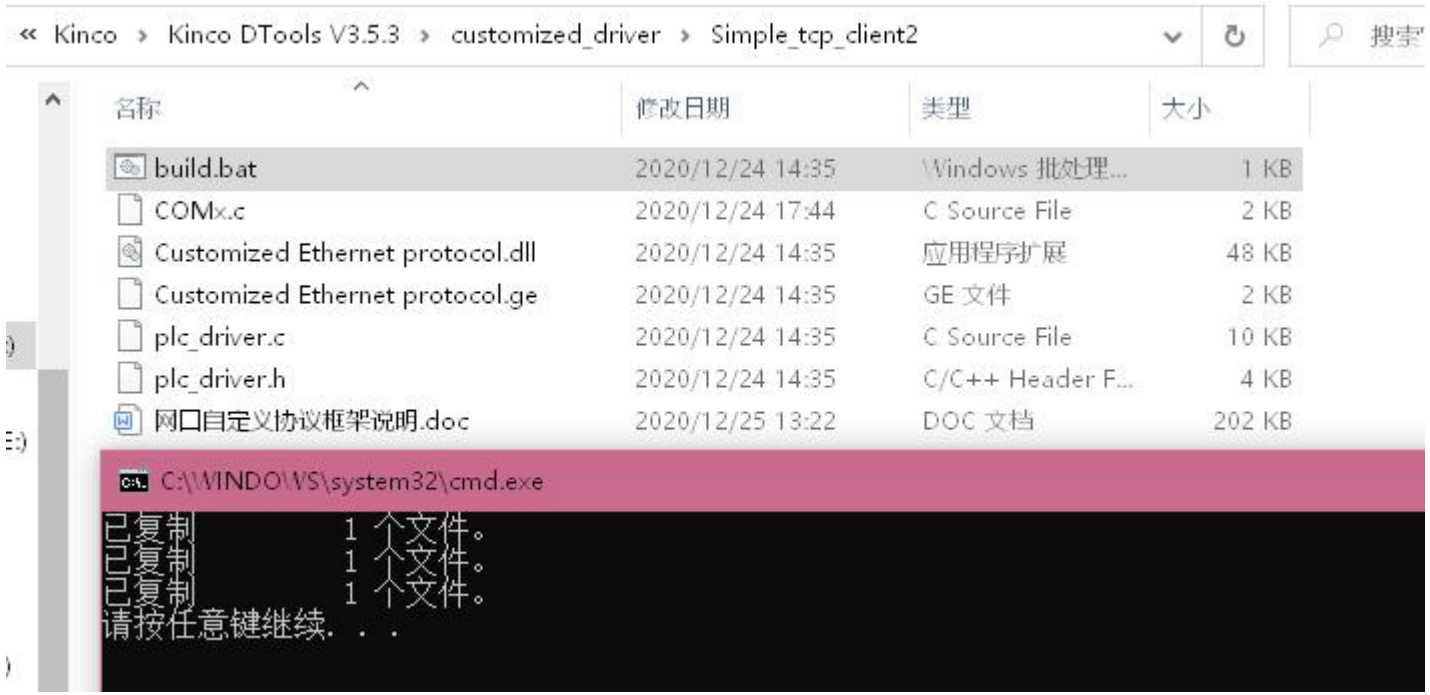
1、在Kinco Dootls安装根目录内新建Customized_driver文件夹



2、将驱动文件包拷贝至新建文件夹内

驱动使用步骤

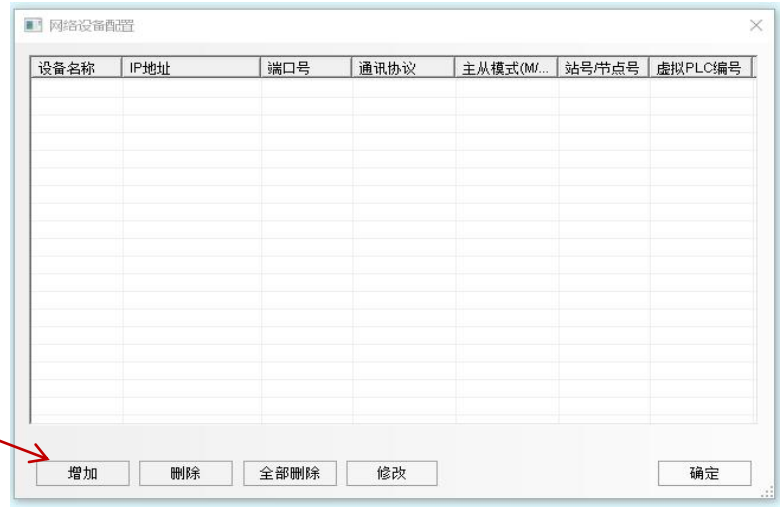
3、打开文件包，双击build后，在没有错误的情况下会自动生成驱动文件。



驱动使用步骤

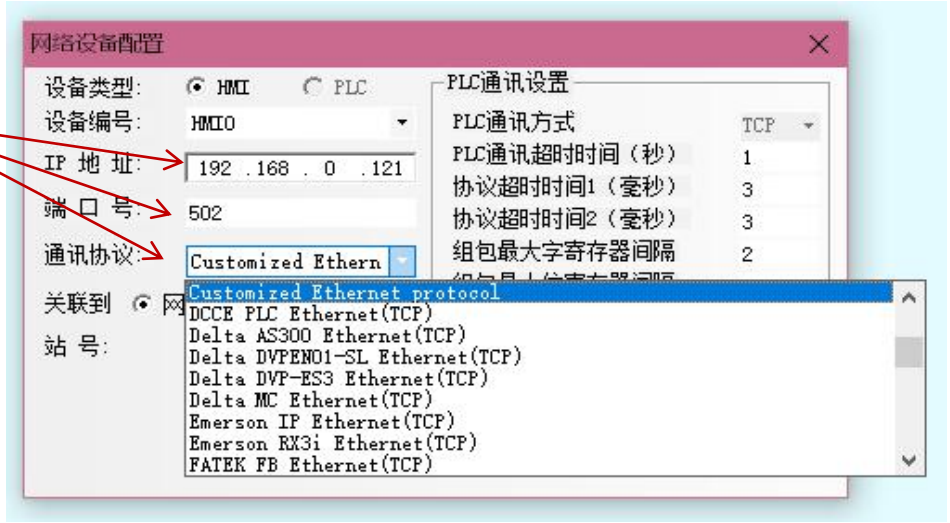
4、打开Kinco Dtools 在通讯协议列表里找到对应的驱动文件，设置通讯参数

1
将以太网拖入组态内



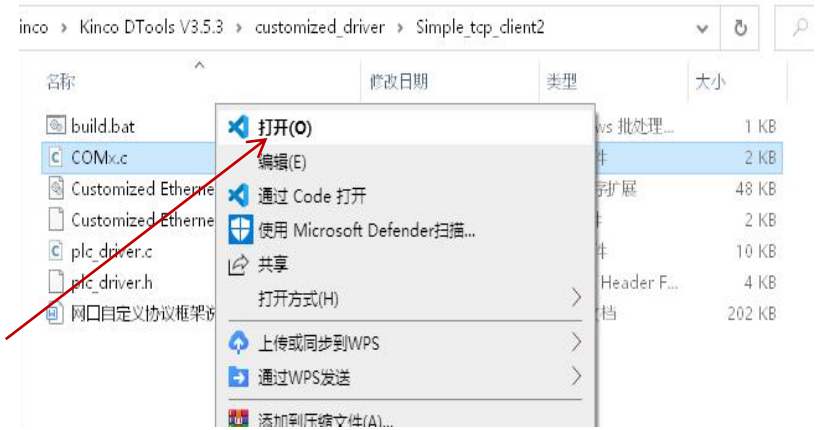
2
点击增加

3
选择自由网口协议，设置通讯的IP地址，以及无占用的通讯端口号（0~65535）



自由协议编写

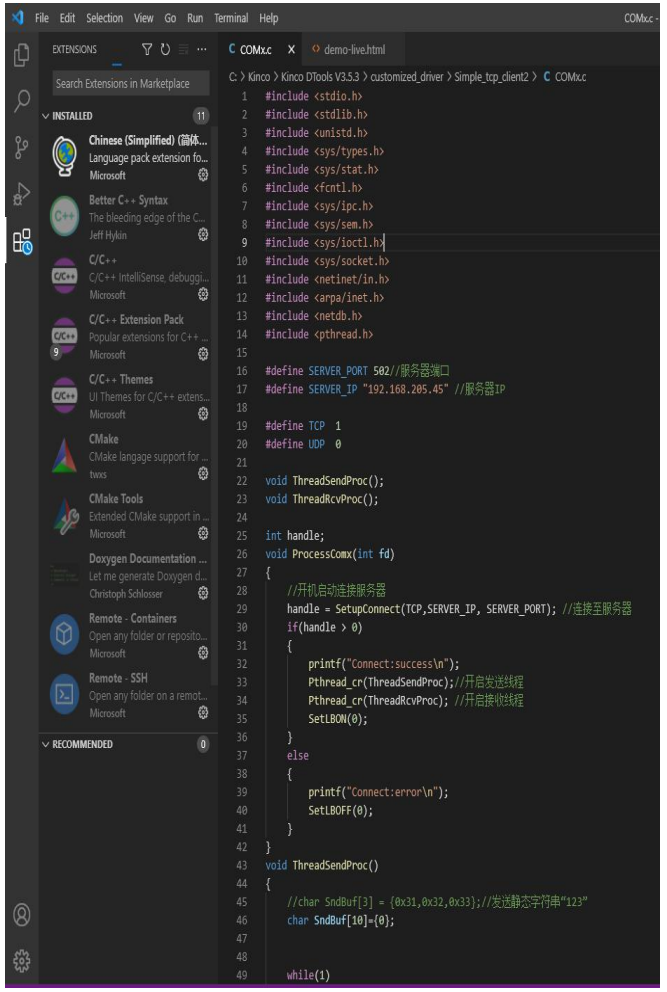
使用记事本或其他编程工具可以打开打开驱动文件夹里面的.C文件，根据实际需求进行编写。



1
右键COMX.C文件

3
编写完成保存后，双击build.bat生成驱动

2
根据需求修改通讯协议

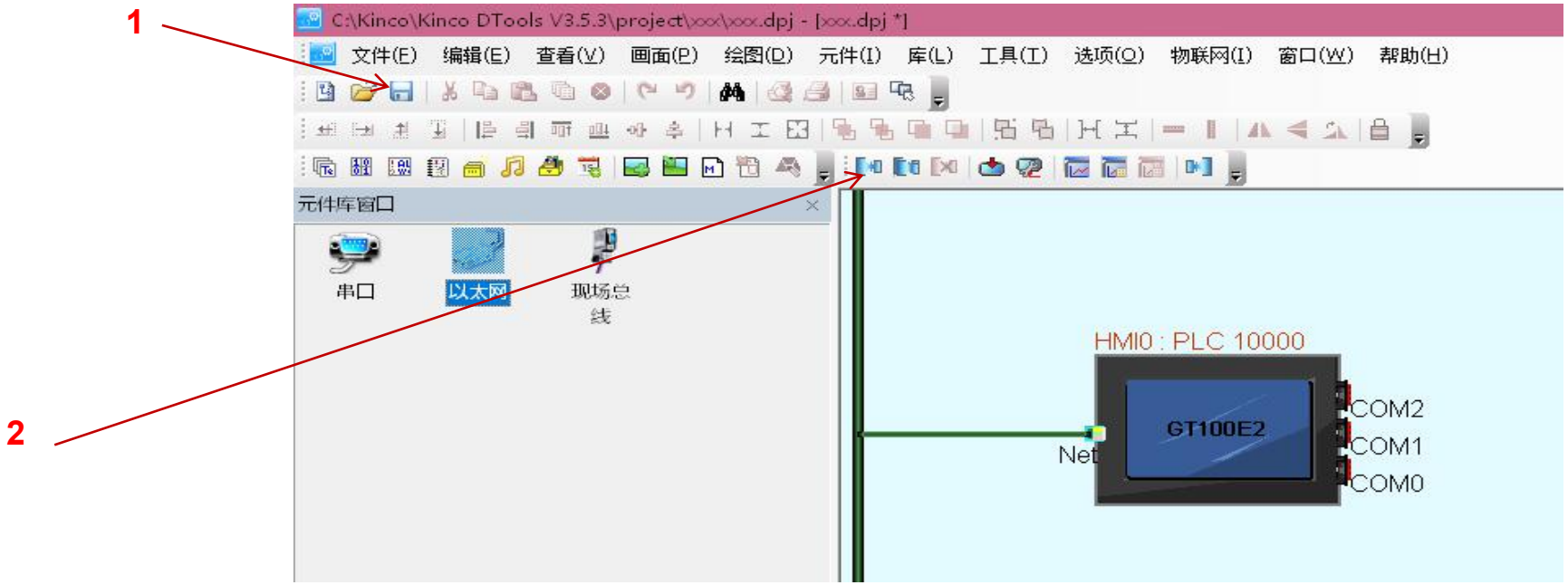


Kinco步科

让中国制造成为全球顶级制造

驱动更新

驱动生成以后打开kinco Dtools ，点击保存，选择全部编译（注意：每次重新生成驱动后都需要重新编译才能生效）



驱动API函数编写

Int SetLRBON (unsigned int addr)
功能：写入RB的寄存器开
参数：addr偏移量

Int SetLRBOFF (unsigned int addr)
功能：写入RB的寄存器关
参数：addr偏移量

int SetupConnect(char type,char *SERVER_IP,int PORT)
功能：网络链接
参数：type通信类型选择TCP=1、UDP=0，SERVER_IP通信IP，PORT端口号

void ev_close_handel(int handle)
功能：关闭链接
参数：handle建立连接的手柄

int Pthread_cr(void thread(void))
功能：创建线程
参数：thread建立的线程函数

驱动API函数编写

void Delay(int ms)
功能：用于延迟ProcessComx的运行周期，防止频繁读写，减轻控制器的通信压力
参数：ms的单位为毫秒, 非零

int ReadData(int fd, unsigned char *read_buff, int count, int ms)
功能：从自由口口读取count个数据，存放在read_buff中。ms是读数据超时时间。
参数：ms的单位为毫秒, 非零。返回值为读取的字符个数，-1表示读超时。

int WriteData(int fd, unsigned char *read_buff, int count)
功能：从自由口口发送count个数据，发送数据存放在read_buff中。

unsigned short Read_LW(unsigned int n)
功能：读取LWn的寄存器的值
参数：n偏移量，最大为9999

void Write_LW(unsigned int n,unsigned short val)
功能：将val 写入LWn的寄存器
参数：n偏移量、最大8999，val待写入的值

void CopyToLW(unsigned int offset,const void *src, int n)
功能：由src所指的内存区域复制n个字节到Lw_offset所在的内存区域
参数：src和LW_offset所在的内存区域不能重叠

驱动API函数编写

void CopyFromLW(unsigned int offset,const void *src, int n)
功能：由Lw_offset所在的内存区域到src所指的内存区域复制n个字节
参数：src和LW_offset所在的内存区域不能重叠

void SetLBON(unsigned int n)
功能：设置LBn寄存器为1
参数：n偏移量、最大8999

void SetLBOFF(unsigned int n)
功能：设置LBn寄存器为0
参数：n偏移量、最大8999

int GetLB(unsigned int n)
功能：读取LBn寄存器的状态
参数：n偏移量、最大9999

int SetLWBON(unsigned int n, unsigned int offset)
功能：设置LW.b寄存器为1,如设置LW5.4，则n=5，offset=4
参数：n偏移量、最大8999，offset最大为16

驱动API函数编写

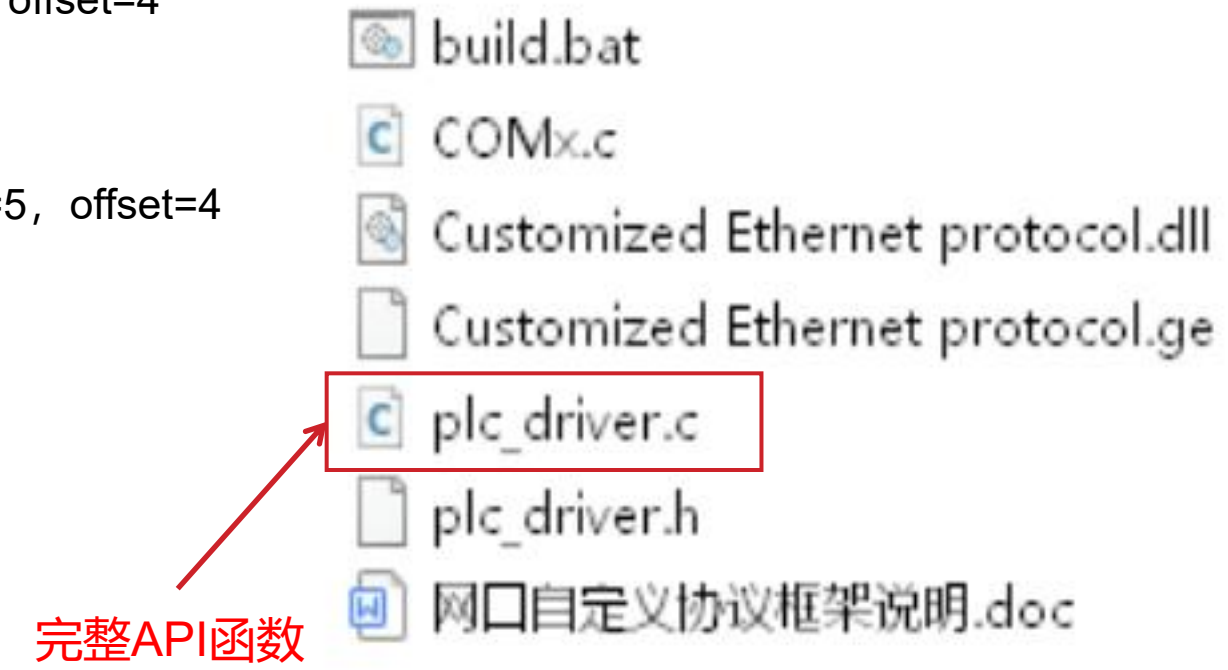
int SetLWBOFF(unsigned int n, unsigned int offset)
功能：设置LW.b寄存器为0,如设置LW5.4，则n=5，offset=4
参数：n偏移量、最大8999，offset最大为16

int GetLWB(unsigned int n, unsigned int offset)
功能：读取LW.b寄存器的状态,如设置LW5.4，则n=5，offset=4
参数：n偏移量、最大9999，offset最大为16

Int Read_Rw(void *buf, int addr, int nWords)
功能：读取RWn的寄存器的值
参数: addr偏移量，nWords 读取个数

Int Write_Rw (void *buf, int addr, int nWords)
功能：写入RWn的寄存器的值
参数: addr偏移量，nWords 写入个数

Int GetRB (unsigned int addr)
功能：读取RB的寄存器的值
参数: addr偏移量



Kinco 步科

代码案例

案例说明

1. 程序使用C语言编写，编辑器为VSCODE。

2. 案例触摸屏型号：GL070E　　案例设备：带网口RFID读卡器

3. 触摸屏做客户端（可以做服务器使用），读卡器做服务器
触摸屏IP：192.168.1.130　　读卡器IP：192.168.1.116
两者IP需要处于同一网段内

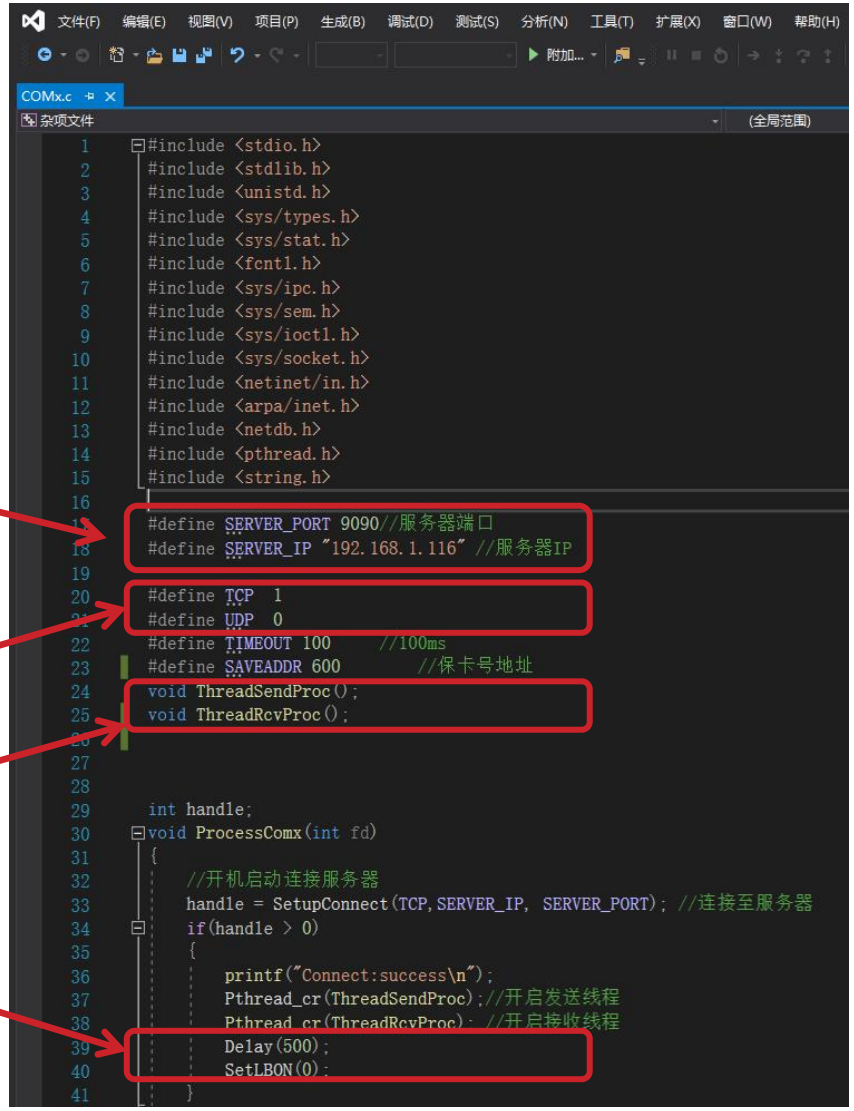
代码说明

9090为服务器端口号
192.168.1.116为服务器ip地址
按需求更改

可以选择通讯协议为TCP或UDP

定义发送报文线程，和读报文线程

若服务器连接成功，LB0置位，定义流程间隔，防止运行过快，导致CPU过载



代码说明

数据发送线程

根据设备手册，查询对应操作的报文，
根据报文格式提前预设

帧头	协议控制字	串行设备地址	数据长度	数据参数	校验码
0xAA	2 字节	1 字节	2 字节(U16)	N 字节	2 字节

帧头：以十六进制“0xAA”作为一帧数据的起始标识。

协议控制字：用于标识当前消息类型。具体位定义如下：

15-14	保留位	保持为 0
13	RS485 标志位	0，此消息不用于 RS485 通信。 1，此消息用于 RS485 通信。
12	读写器主动上传消息标志位	0，表示此消息是上位机指令或者读写器对上位机指令的响应，不是读写器主动上传信息。 1，表示此消息为读写器主动上传的消息。
11-8	消息类别号	0，读写器错误或告警消息。 1，读写器配置与管理消息。 2，RFID 配置与操作消息。 3，读写器日志消息。 4，读写器应用处理器软件与基带软件升级消息。 5，测试指令 6，定制扩展消息 0x7~0x8，保留。 0x9，国网 RFID 外设。 0xA，扩展 MCU 消息。 0xB~0xF，保留。
7-0	消息 ID	0x00~0xFF，区分同一类别消息下的具体消息，下文中简称 MID。

```
42      else
43      {
44          printf("Connect:error\n");
45          SetLBOFF(0);
46      }
47  }
48  void ThreadSendProc()
49  {
50      //char SndBuf[3] = {0x31,0x32,0x33};//发送静态字符串“123”
51      char SndBuf[10]={0};
52      unsigned char send[17] = { 0xAA, 0x02, 0x11, 0x00, 0x0A, 0x01, 0x01, 0x00, 0x02, 0x00, 0x04,
53                                 0x00, 0x00, 0x00, 0x01, 0xE0, 0xA1 };
54      unsigned char stopCMD[7] = { 0xAA, 0x02, 0xFF, 0x00, 0x00, 0xA4, 0x0F };
55      //AA02100002010171AD
56      unsigned char startCMD[9] = { 0xAA, 0x02, 0x10, 0x00, 0x02, 0x01, 0x01, 0x71, 0xAD };
57      unsigned char powerSet[9] = { 0xAA, 0x02, 0x01, 0x00, 0x02, 0x01, 0x1B, 0xF6, 0x0A };
58
59      while(1)
60      {
61          if(GetLB(1000))//LB1000发送报文
62          {
63              CopyFromLW(5000,SndBuf, 10);//发送动态LW5000-LW5004的内容
64              WriteData(handle, SndBuf, 10);
65              SetLBOFF(1000);
66
67          }
68
69      }
70  }
71
```

LB1000开控制报文发送，发送一次后自动关闭。
该函数专门用于报文发送，修改SndBuf的内容，
或直接替换为预设的报文，即可发送对应报文

代码说明

数据接收线程

定义数据接受区，数据缓存区

data为收到的字节数

将收到的数据长度已经数据内容存入LW寄存器

```
void ThreadRcvProc()
{
    char buff[10];
    char receive[256] = "";
    int data = 0;

    while (1)
    {
        int i = 0;
        data = 0;
        int status;

        data = ReadData(handle, (unsigned char*)&receive, 256, TIMEOUT);
        if(data > 0)
        {
            SetLBON(998);
            printf("receive_num:%d\n", data);
            Write_LW(0, data);
            Write_LW(1, receive[0]);
            Write_LW(2, receive[1]);
            Write_LW(3, receive[2]);
            Write_LW(4, receive[3]);
            Write_LW(5, receive[4]);
            Write_LW(6, receive[5]);
            Write_LW(7, receive[6]);
            Write_LW(8, receive[7]);
            Write_LW(9, receive[8]);
            Write_LW(10, receive[9]);
            Write_LW(11, receive[10]);
            Write_LW(12, receive[11]);
            Write_LW(13, receive[12]);
            Write_LW(14, receive[13]);
            Write_LW(15, receive[14]);
            Write_LW(16, receive[15]);
            Write_LW(17, receive[16]);
            Write_LW(18, receive[17]);
            Write_LW(19, receive[18]);
            Write_LW(20, receive[19]);
            Write_LW(21, receive[20]);
```

代码说明

数据接收解析

定义自增i，轮询数据接收区，寻找与卡号数据

将卡号存入数据缓存区

将数据缓存区内数据传递至LW在触摸屏上显示

```
106         Write_LW(17, receive[16]);
107         Write_LW(18, receive[17]);
108         Write_LW(19, receive[18]);
109         Write_LW(20, receive[19]);
110         Write_LW(21, receive[20]);
111         Write_LW(22, receive[21]);
112         if (data > 0)
113         {
114             status = 2;
115             for (i = 0; i < data; i++) {
116                 //AA 12 00 00 14 00 04 //AA 12 00 00 14 00 06
117                 //printf("%02x ", receive[i]);
118                 if (receive[i] == 0XAA && receive[i + 1] == 0X12 && receive[i + 2] == 0X00 &&
119                     receive[i + 3] == 0X00 && receive[i + 4] == 0X14 && receive[i + 5] == 0X00 && receive[i + 6] == 0X04)
120                 {
121                     memcpy(buff, (const void*)&receive[i + 7], 4);
122                     status = 1;
123                 }
124             }
125
126             printf("\r\n");
127             //Delay(500);
128         }
129         if (status == 1)
130         {
131             printf("success: ");
132             for (i = 0; i < 10; i++)
133             {
134                 Write_LW(SAVEADDR + i, buff[i]);
135                 printf("%02X ", buff[i]);
136             }
137         }
138         else
139         {
140             for (i = 0; i < 10; i++)
141             {
142                 Write_LW(SAVEADDR + i, 0);
143             }
144             printf("filed: %d\r\n", status);
145         }
146     }
```

Kinco步科

PC与HMI通讯案例

案例说明

1. 本案例使用WIN10系统，案例使用软件：NetAssist、Wireshark

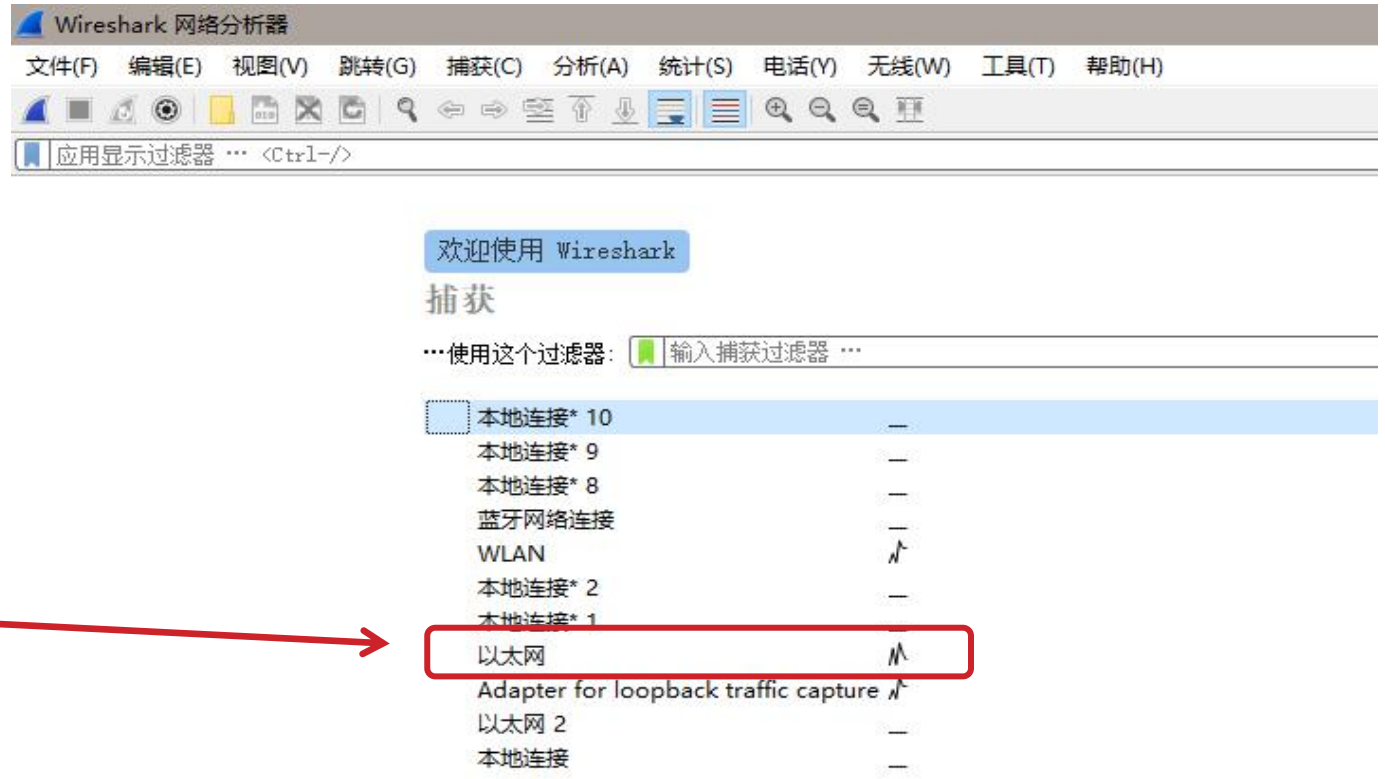
2. 案例触摸屏型号：GL070E　　案例设备：PC

3. 触摸屏做客户端，连接PC，实现数据互传

4. 根据之前的代码案例，修改电脑IP地址为：192.168.1.116

案例说明

打开WrieShark，选择监控以太网

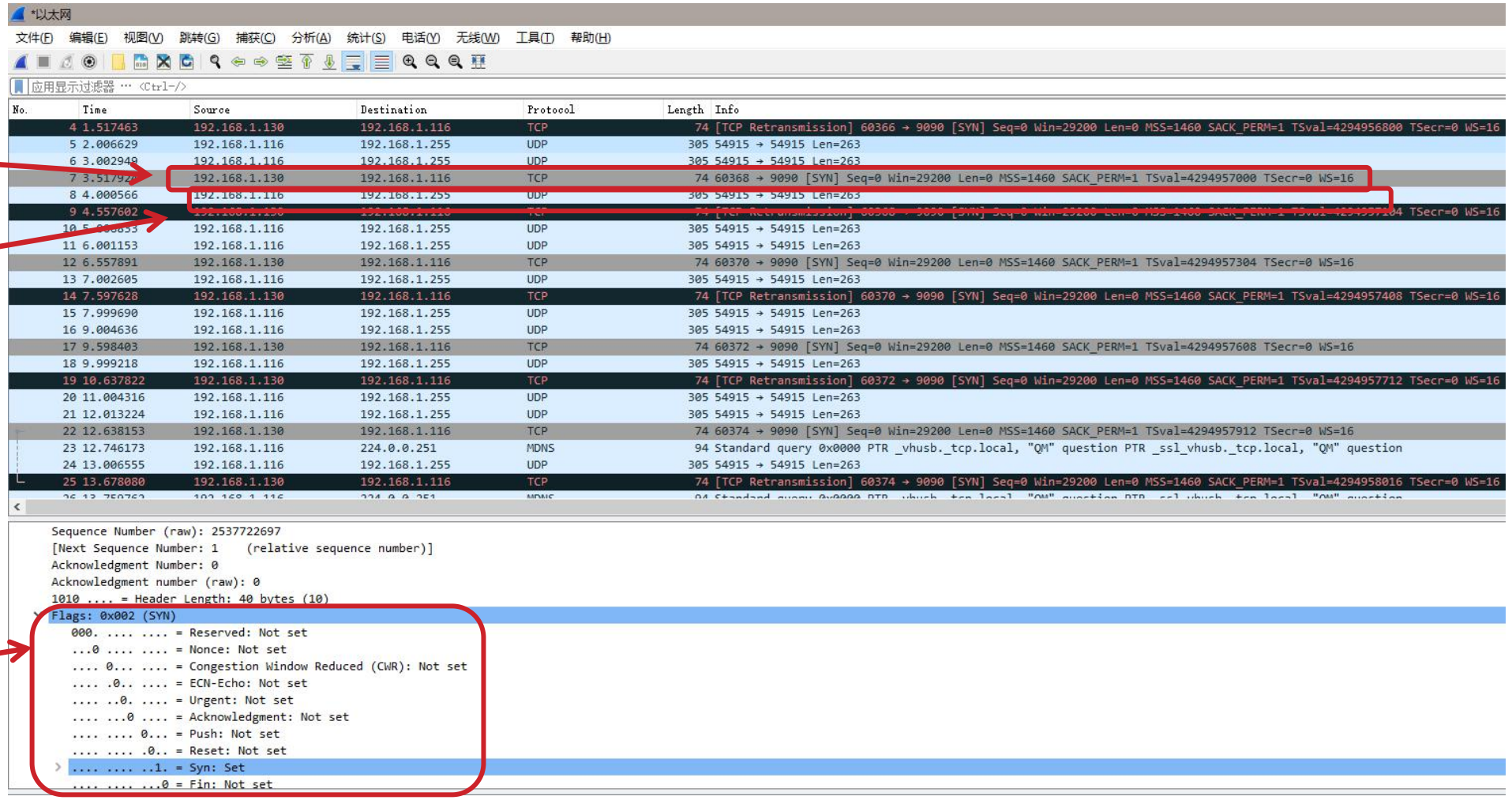


案例说明

触摸屏一直在向PC发送握手请求

由于未开启TCP Server，无法回应对应请求，超过延时，产生黑包，若已开启，请关闭防火墙重试

标志位



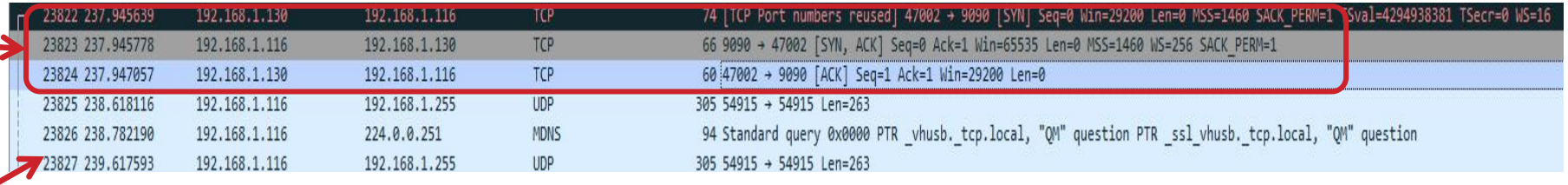
案例说明

打开NetAssist，设置协议为TCP Server，选择以太网地址，端口号与代码一致9090

设置完成点击打开

案例说明

触摸屏一直在向
与PC握手完成

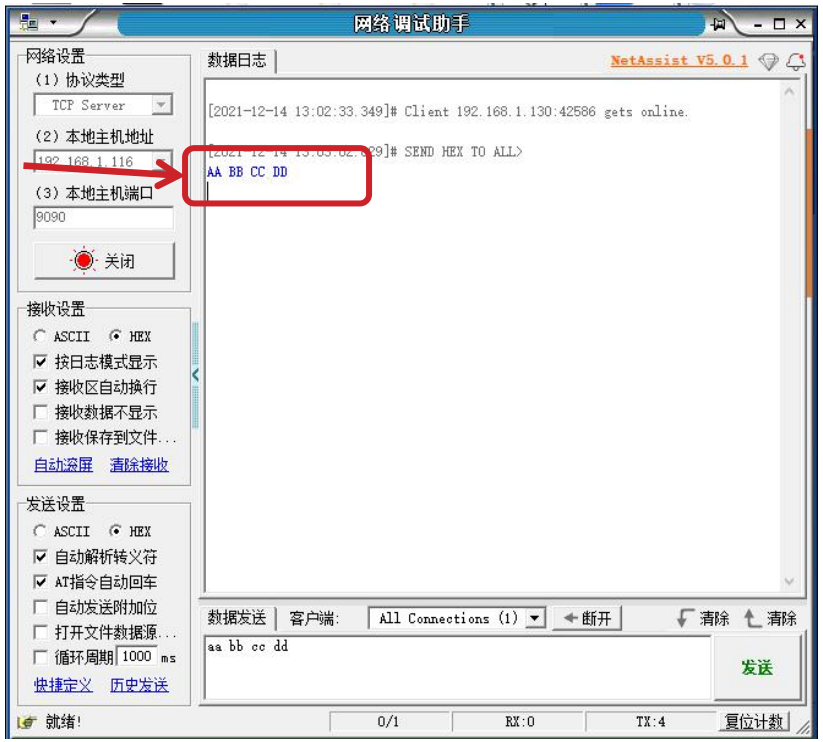


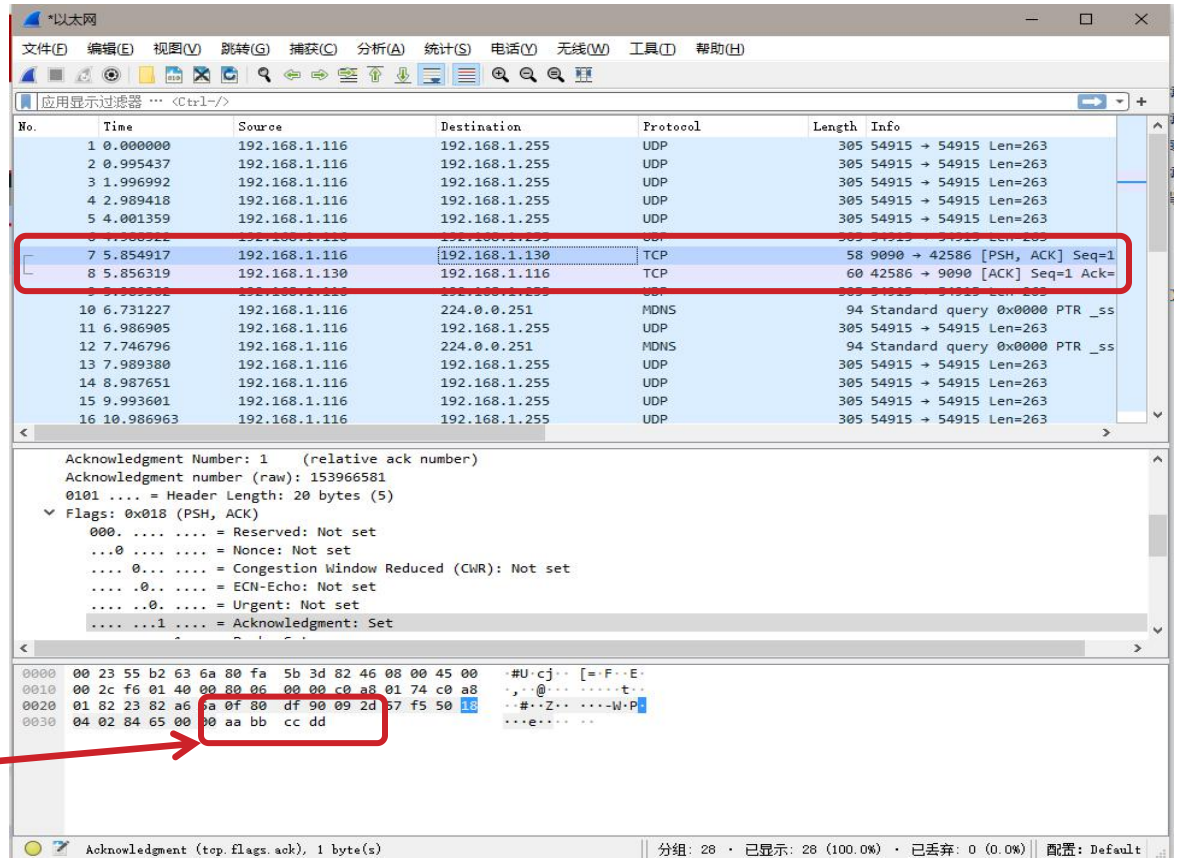
23822 237.945639	192.168.1.130	192.168.1.116	TCP	74	[TCP Port numbers reused] 47002 → 9090 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4294938381 TSecr=0 WS=16
23823 237.945778	192.168.1.116	192.168.1.130	TCP	66	9090 → 47002 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
23824 237.947057	192.168.1.130	192.168.1.116	TCP	60	47002 → 9090 [ACK] Seq=1 Ack=1 Win=29200 Len=0
23825 238.618116	192.168.1.116	192.168.1.255	UDP	305	54915 → 54915 Len=263
23826 238.782190	192.168.1.116	224.0.0.251	MDNS	94	Standard query 0x0000 PTR _vhusb._tcp.local, "QM" question PTR _ssl_vhusb._tcp.local, "QM" question
23827 239.617593	192.168.1.116	192.168.1.255	UDP	305	54915 → 54915 Len=263

可以开始数据交互

案例说明

PC向触摸屏发送报文：AA BB CC DD





数据成功发送，触摸屏返回ACK帧，确认收到

Kinco 步科

三、使用注意事项

使用注意事项

1. 编写驱动使用C语言进行编程，编译出错会在运行build.bat时提示，出错则不能成功生成驱动。

2. 只要是更改了程序，必须重新build生成驱动。

3. 只要是重新生成了驱动，在组态软件里必须执行全部编译操作，驱动的更改才能生效。

Kinco 步科

谢谢观看

2021