

Kinco步科

自由口串口通讯

2021



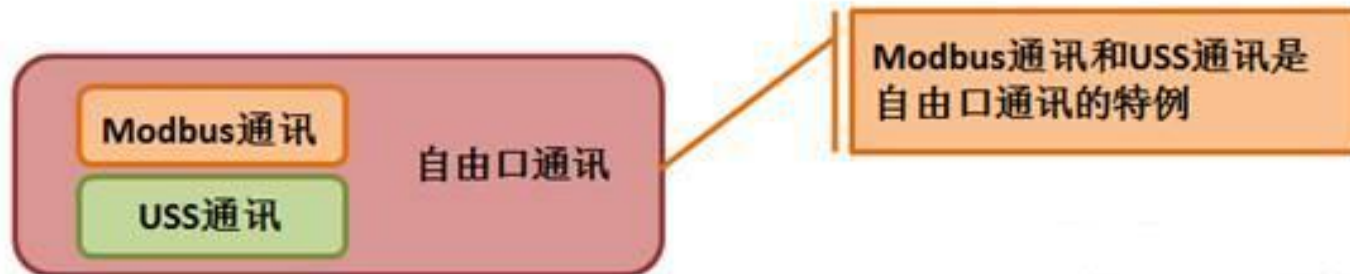
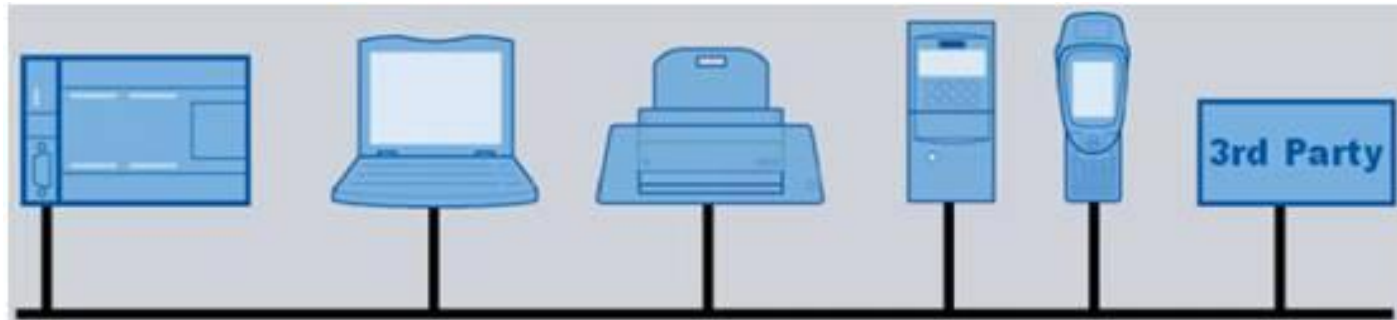
- 1 自由口串口的定义以及应用场景
- 2 步科自由口使用说明
- 3 使用注意事项

一、自由口通讯的定义 以及应用场景

自由口通讯的定义

主要用于两台及以上设备通讯时，没有共同的通讯协议，就需要使用自由口通讯，自由口通讯也叫无协议通讯。

当使用串口进行自由口通讯时，也需要匹配接口形式（232 485）、波特率、数据位、校验位停止位 以及报文格式等，西门子USS通讯以及我们通常使用的MODBUS-RTU也是一种特定形式的自由口通讯。



自由口通讯的应用场景

1

物料扫码枪以及RFID物料扫码系统

2

没有固定通讯协议的仪表，外围设备通讯

3

需要对数据进行一定量的解析，自定义数据格式の場合

Kinco步科

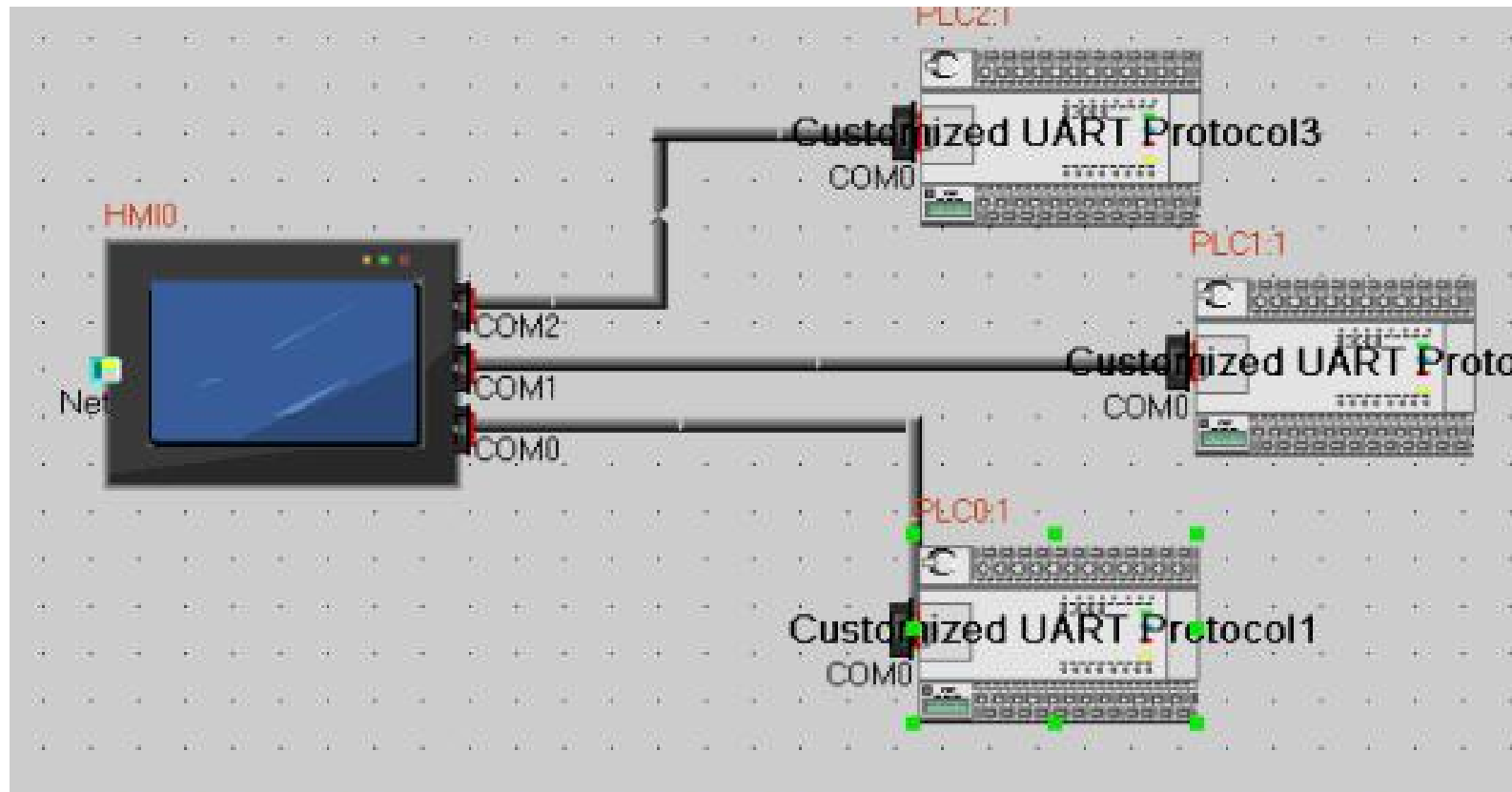
二、使用说明

Kinco步科

让中国制造成为全球顶级制造

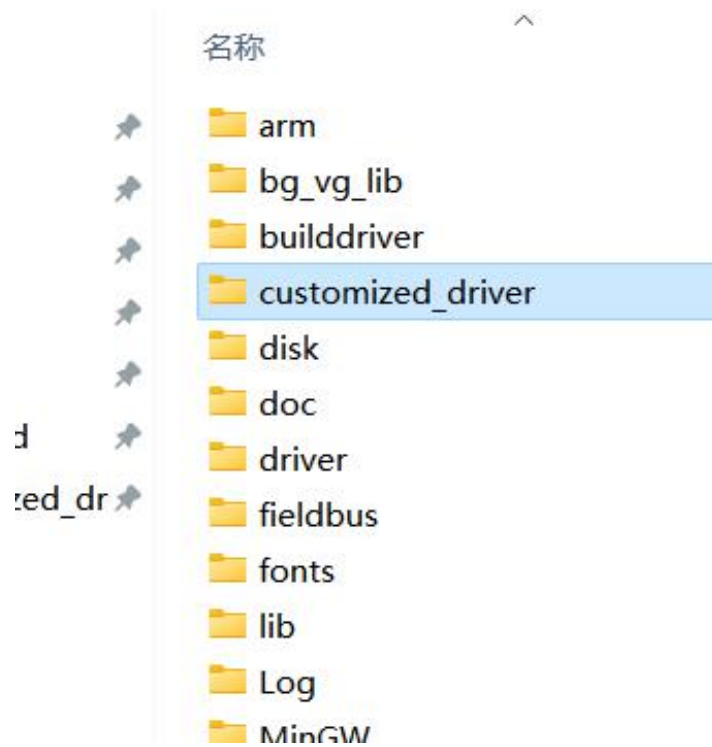
步科自由口使用说明

串口自定义协议的驱动目前可以支持三个串口同时定制自由协议，分别为Customized UART protocol1, Customized UART protocol2, Customized UART protocol3, 可对应三个串口。组态的使用方式同其他的PLC组态方式一样，直接连接屏即可，通过LW和LB寄存器进行数据的交换。

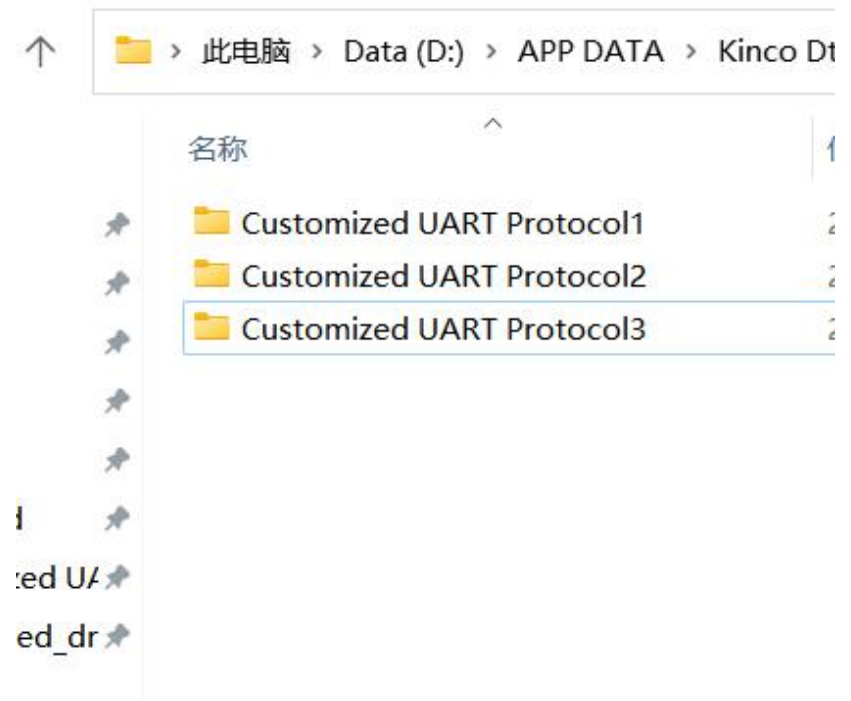


驱动使用步骤

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



2、将驱动文件包拷贝至新建文件夹内



Kinco 步科

让中国制造成为全球顶级制造

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，需要使用哪个串口就在对应的驱动包里点击bulid, 生成哪个串口的驱动

The screenshot shows a file explorer window with a list of files. A red arrow labeled '1' points to the 'build' file. Below the file list, a command prompt window is open, showing the execution of the 'build' file. A red arrow labeled '2' points to the command prompt window.

名称	修改日期	类型	大小
build	2018-03-08 上午 11:18	Windows 批处理...	1 KB
COMx.c	2016-04-17 下午 9:55	C Source	7 KB
COMx-ind205.c	2016-03-15 下午 2:26	C Source	5 KB
COMx-xk.c	2016-03-15 下午 2:26	C Source	5 KB
COMx-串口键盘.c	2016-04-06 下午 4:28	C Source	3 KB
Customized UART protocol1.dll	2013-05-14 下午 1:28	应用程序扩展	48 KB
Customized UART protocol1.ge	2013-05-14 下午 1:28	GE 文件	2 KB
drv_p0.so	2021-12-10 下午 4:29	SO 文件	17 KB
libserail.a	2018-03-08 上午 11:15	A 文件	9 KB

```
C:\WINDOWS\system32\cmd.exe
已复制      1 个文件。
已复制      1 个文件。
已复制      1 个文件。
请按任意键继续. . .
```

驱动使用步骤

4、打开Kinco Dtools 在PLC列表里找到对应的驱动文件，拖入组态即可，设置通讯参数

The image illustrates the steps for configuring a custom UART protocol driver in Kinco Dtools. It is divided into three numbered sections:

- 1:** Selection of the driver file. In the left-hand tree view, 'Customized UART Protocol1' is highlighted under the 'HMI/属性' (HMI/Properties) category.
- 2:** Dragging the driver into the configuration area. A red arrow points to the 'Customized UART Protocol1' icon being placed on the 'COM1' port of the 'GT070E-WiFi' HMI device.
- 3:** Configuration of communication parameters. A dialog box titled 'HMI/属性' is shown with the following settings:
 - 通讯类型 (Communication Type): RS232
 - 波特率 (Baud Rate): 9600
 - 数据位 (Data Bits): 8
 - 奇偶校验 (Parity): 无校验 (None)
 - 停止位 (Stop Bits): 1
 - 广播站号 (Broadcast Station Number): 65535A '高级设置' (Advanced Settings) button is located at the bottom of the dialog.

The right side of the image shows a schematic diagram of the HMI (GT070E-WiFi) connected to a PLC (PLC_0_1) via a 'Customized UART Protocol1' connection line.

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：从串口读取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：从串口发送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

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

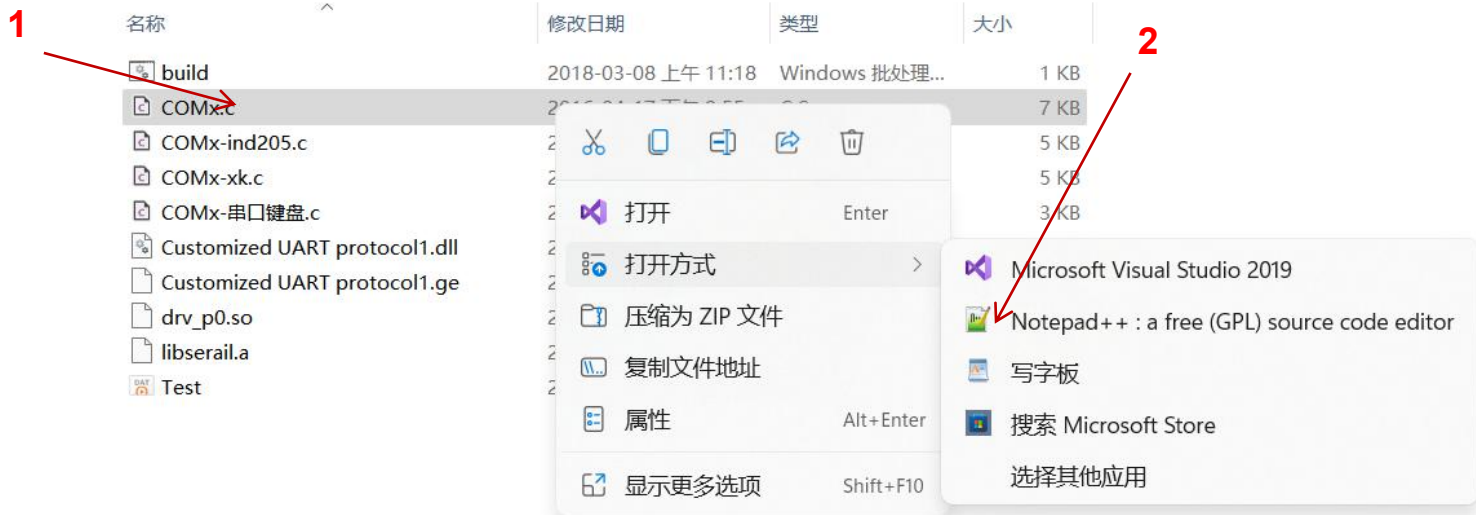
int GetLWB(unsigned int n, unsigned int offset)

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

驱动API函数编写

使用Notepad++打开驱动文件夹里面的.C文件根据实际需求进行编写。



```
73  
74  
75  
76  
77 void ProcessComx(int fd)  
78 {  
79     unsigned char receive[1024];  
80     unsigned char send[1024];  
81     int i,timeout,crc_result;  
82     unsigned short LwValue;  
83     handle=fd;  
84  
85     Pthread_cr(thread);//线程  
86  
87     while(1)  
88     {  
89         Delay(2500);//ms  
90         Write_Rw( receive, 0, 1);  
91         printf("COMx Working...\r\n");  
92         if (GetLB(0)==1)  
93         {  
94             LwValue=Read_LW(3);  
95             if(GetLB(1)==1)  
96                 SetLWBOFF(1);  
97             else  
98                 SetLBON(1);  
99  
100             if(GetLWB(4, 5)==1)//lw4.5  
101                 SetLWBOFF(4,5);  
102             else  
103                 SetLWBON(4,5);  
104  
105             printf("LW3=%d\r\n", LwValue);  
106             send[0] = 1;  
107             send[1] = 3;  
108             send[2] = 0;  
109             send[3] = 0;  
110             send[4] = 0;  
111             send[5] = 2;  
112             send[6] = (unsigned char)(crc_result>>8);  
113             send[7] = (unsigned char)crc_result;  
114             crc_result=crc16(send,6);  
115             WriteData(fd, send, 8);  
116  
117             timeout = 100;//ms  
118             ReadData(fd, receive, 1024, timeout);  
119             //for(i=0;i<10;i++)  
120             // printf("%d ", receive[i]);  
121             Write_LW(0, receive[3]*0x100+receive[4]);  
122             Write_LW(1, receive[5]*0x100+receive[6]);  
123         }  
124     }  
125  
126 }
```

3

步科

让中国制造成为全球顶级制造

驱动API函数编写

编写完成后保存，保存完成后双击build, 生成驱动文件

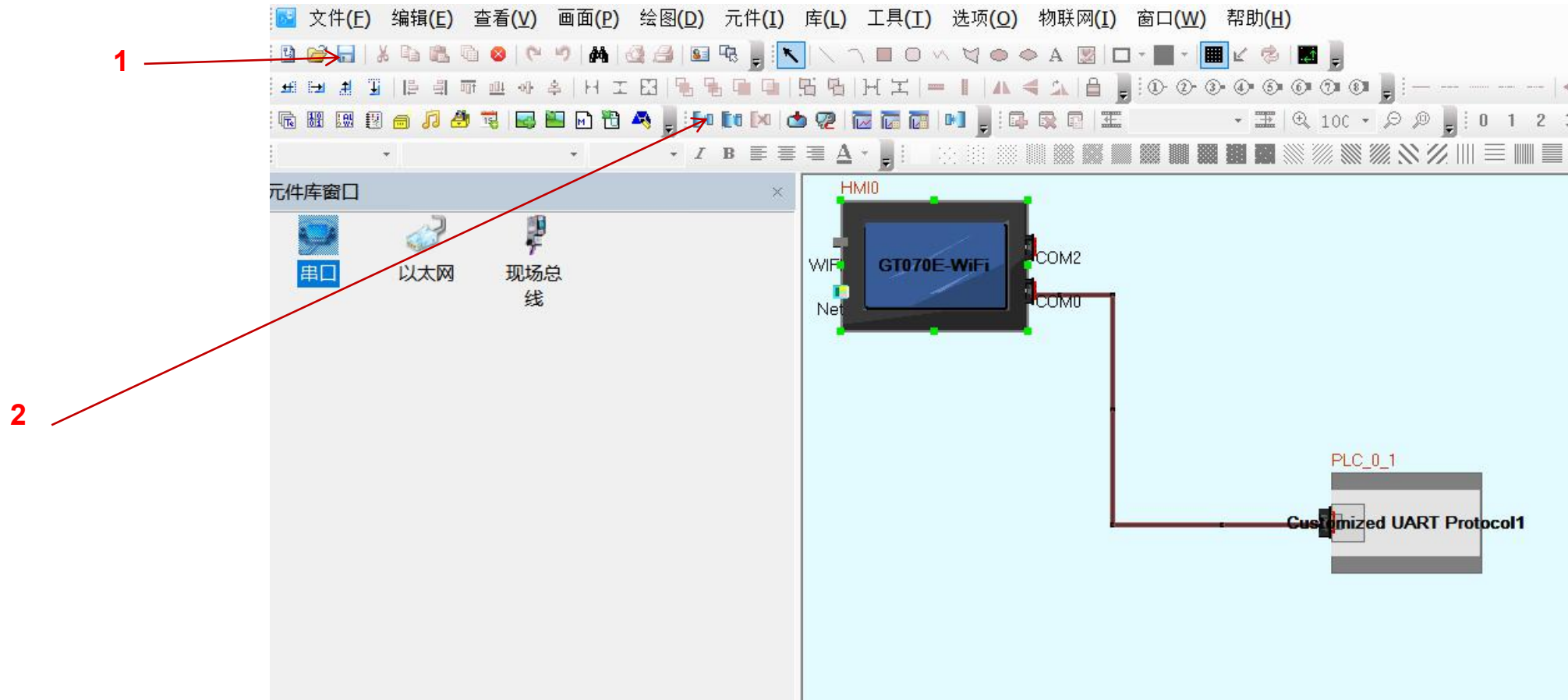
The screenshot shows a Windows File Explorer window with a table of files. A red arrow labeled '1' points to the 'build' file. A second red arrow labeled '2' points to a terminal window showing the execution of 'cmd.exe' with the output '已复制 1 个文件。' repeated three times.

名称	修改日期	类型	大小
build	2018-03-08 上午 11:18	Windows 批处理...	1 KB
COMx.c	2016-04-17 下午 9:55	C Source	7 KB
COMx-ind205.c	2016-03-15 下午 2:26	C Source	5 KB
COMx-xk.c	2016-03-15 下午 2:26	C Source	5 KB
COMx-串口键盘.c	2016-04-06 下午 4:28	C Source	3 KB
Customized UART protocol1.dll	2013-05-14 下午 1:28	应用程序扩展	48 KB
Customized UART protocol1.ge	2013-05-14 下午 1:28	GE 文件	2 KB
drv_p0.so	2021-12-10 下午 4:29	SO 文件	17 KB
libserail.a	2018-03-08 上午 11:15	A 文件	9 KB

```
C:\WINDOWS\system32\cmd.exe
已复制 1 个文件。
已复制 1 个文件。
已复制 1 个文件。
请按任意键继续. . .
```

驱动API函数编写

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)



无协议通讯

Universal HEX Slave

Device	Bit Address	Word Address	Format	Notes
send	LB 0	-----	D	对应 COM0 通讯 数据
clear	LB1	-----	D	
Send data buffer	-----	LW 0~99	DD	
Receive data buffer	-----	LW 100~199	DDD	
send	LB2	-----	D	对应 COM1 通讯 数据
clear	LB3	-----	D	
Send data buffer	-----	IW 200~299	DD	

无协议通讯

Receive data buffer	-----	LW 300~399	DDD	
send	LB 4	-----	D	对应 COM2 通讯 数据
clear	LB5	-----	D	
Send data buffer	-----	LW400~499	DD	
Receive data buffer	-----	LW 500~599	DDD	
send	LB6	-----	D	对应 COM3 通讯 数据
clear	LB7	-----	D	
Send data buffer	-----	LW600~699	DD	
Receive data buffer	-----	LW 700~799	DDD	

Kinco步科

二、使用注意事项

Kinco步科

让中国制造成为全球顶级制造

使用注意事项

1

驱动文件包分别对应触摸屏COM1 COM2 COM3,使用对应驱动包必须对应使用对应端口

2

编写驱动时，不能出现语法错误，否则不能成功生成驱动。

3

只要是更改了驱动程序，必须重新生成驱动才能生效。

4

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

