

ProU C 语言 TCP 通讯方式

简单编程，提升效率



做专业的装备开发平台



目录

1 ProU TCP 通讯-C 语言方式（以太网） 3

1.1 ProU 的设置 3

1.2 C 程序的编写 4

1.3 设置变量 10

1.4 编写梯形图 12

1.5 制作可视化界面 13

1.6 通讯测试 17

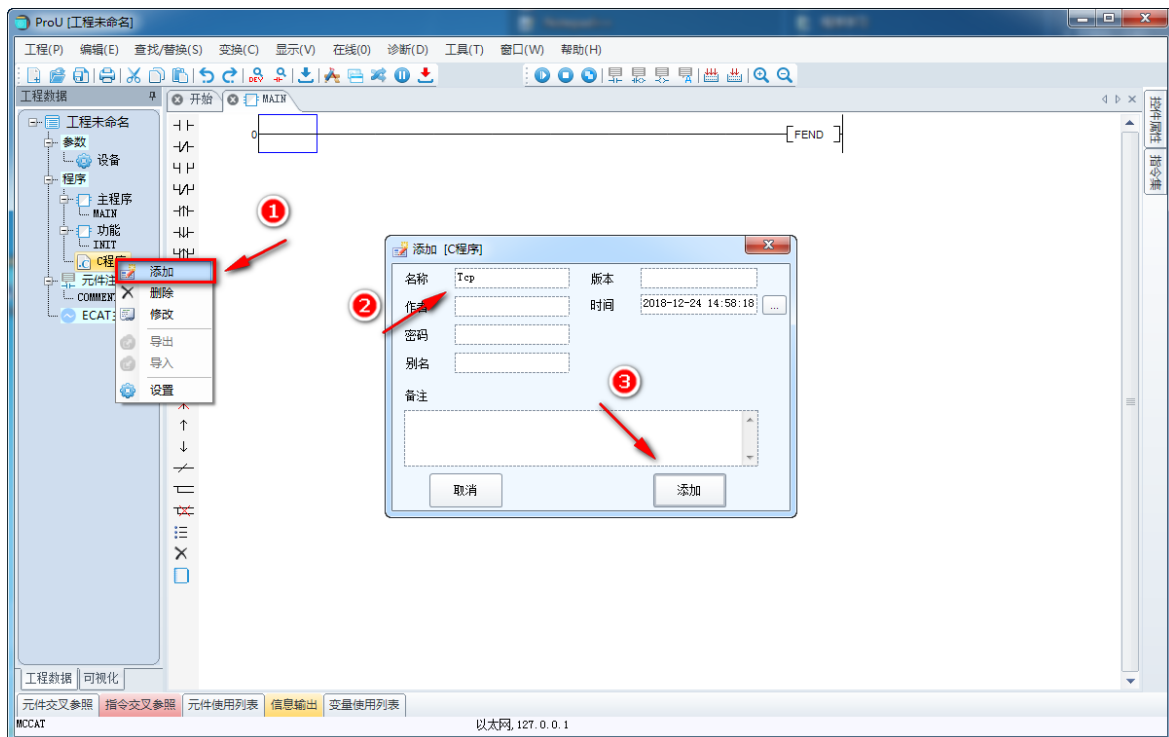
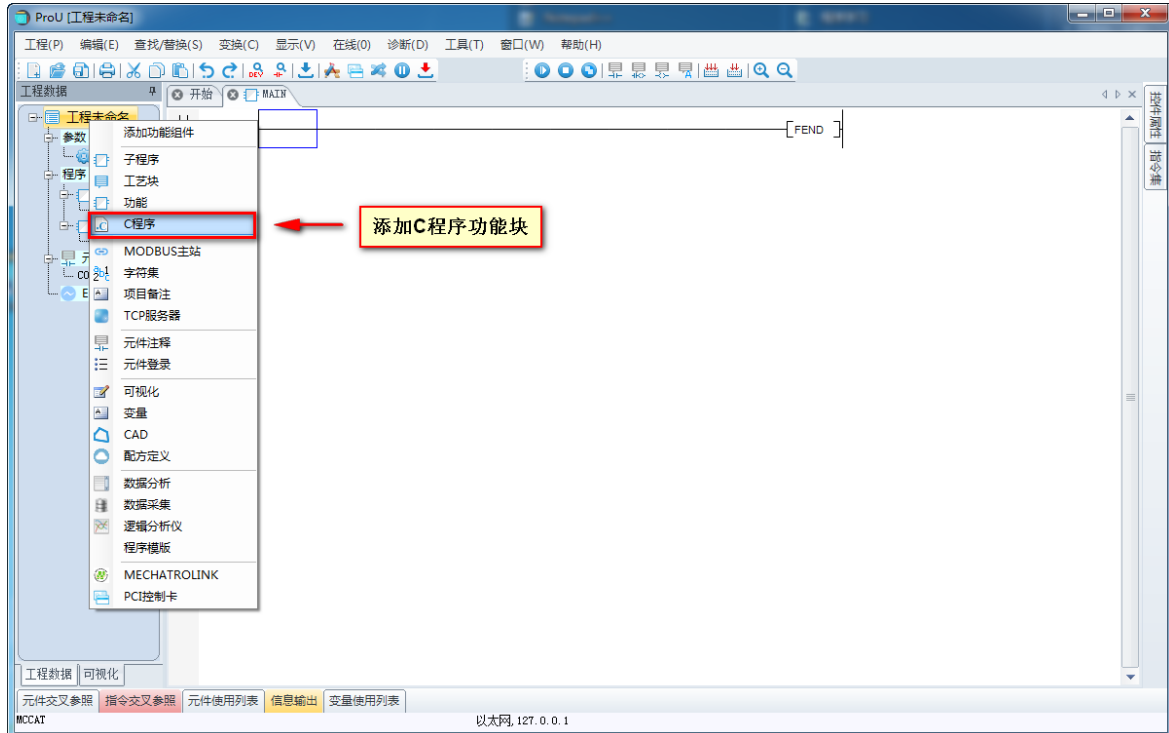




1 ProU TCP 通讯-C 语言方式（以太网）

1.1 ProU 的设置

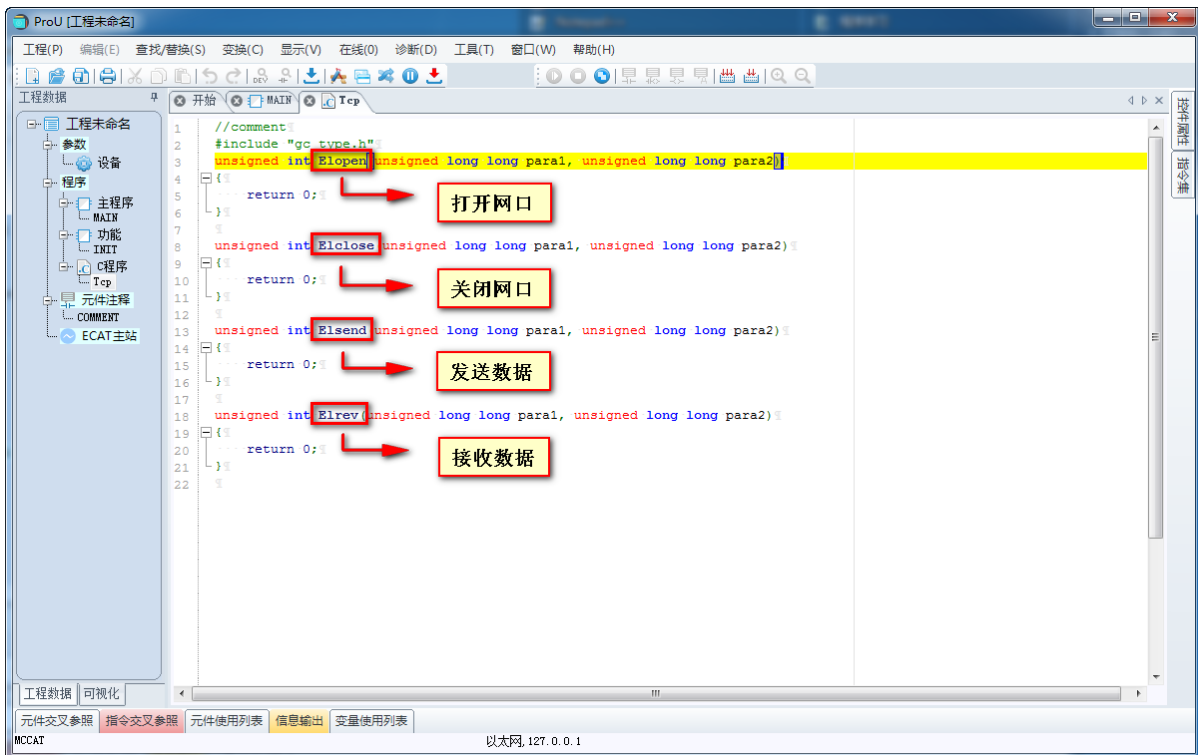
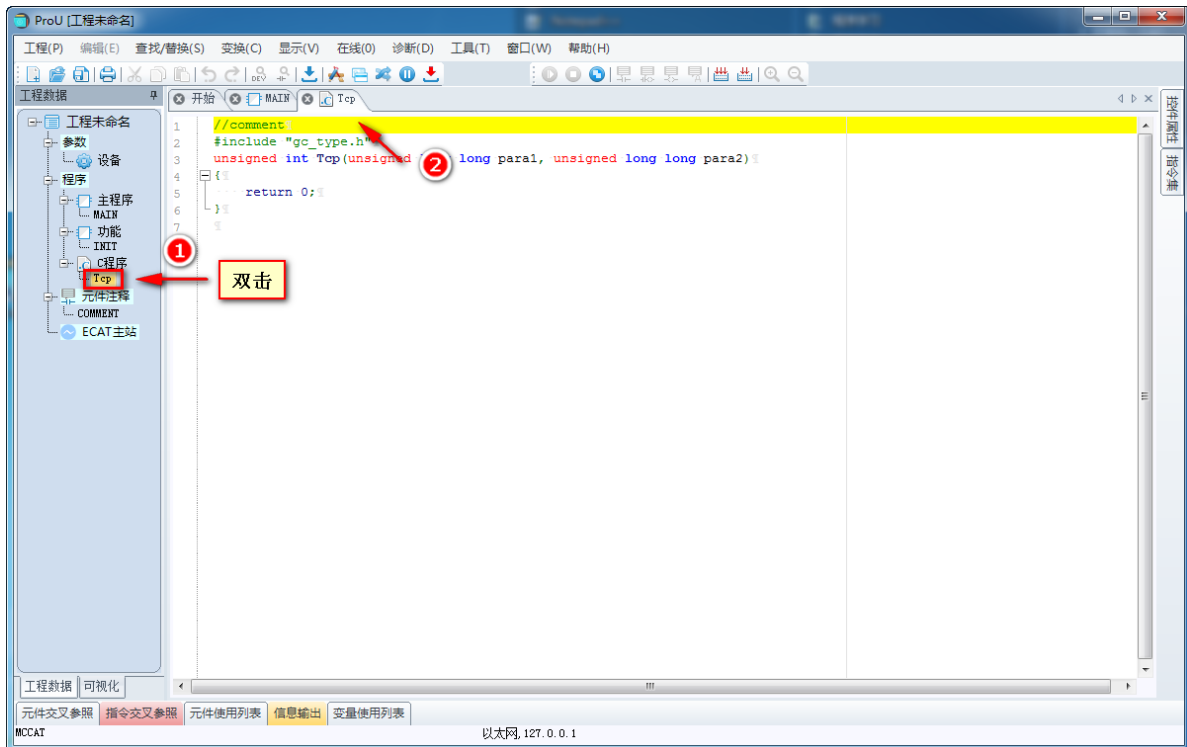
1. 在 ProU 中新建项目工程程序后，在左侧项目数据中添加 C 程序功能块，再添加 C 程序。





1.2 C 程序的编写

1、进入新建好的 C 程序中。因为 TCP 通讯需要打开网口-关闭网口-发送数据-接收数据四个功能，所以可以分别建立四段程序，并分别命名。

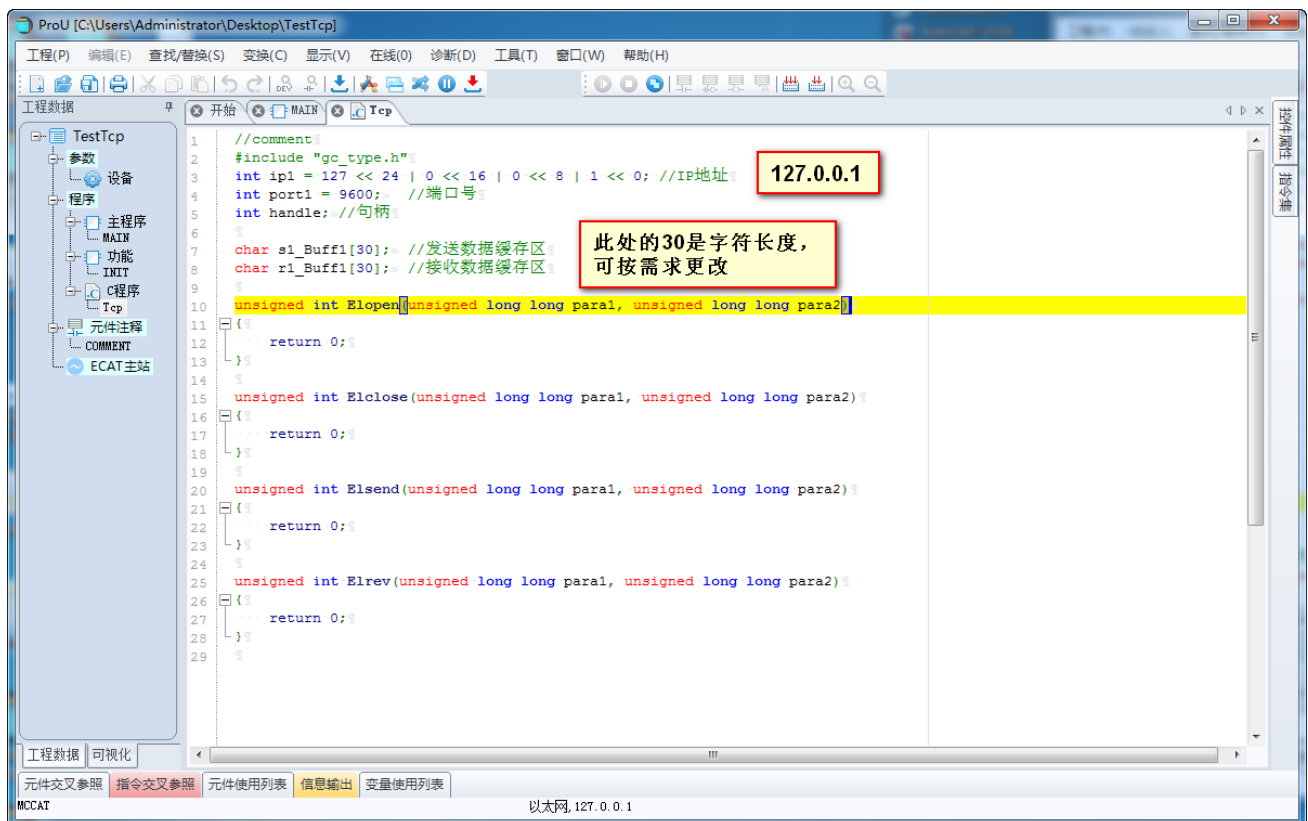




2、因为在 ProU 的 C 程序中有专门用于 TCP 通讯的功能语句。所以需要在开头声明全局变量，IP 地址、端口号、句柄、接收/发送数据缓存区等，以便在程序中使用。

<code>intGC_Eth_Open(intip, intport)</code>	打开网口 ip:IP地址， 如配置127.0.0.1 ip_addr = 127 << 24; ip_addr += 0 << 16; ip_addr += 0 << 8; ip_addr += 1 << 0; port:端口号 返回打开的句柄
<code>intGC_Eth_Close(inthandle)</code>	关闭网口 handle:打开是返回的句柄 返回值:>=0 关闭成功 <0 关闭失败
<code>intGC_Eth_Send(inthandle, char *buffer, intlen)</code>	网口发送 handle:打开是返回的句柄 buffer:发送缓冲区 len:发送字节数 返回值:发送的字节数
<code>intGC_Eth_Rev(inthandle, char *buffer, intlen)</code>	网口接收 handle:打开是返回的句柄 buffer:接收缓冲区 len:接收字节数 返回值:接收的字节数

(TCP 网口通讯功能语句，若想查看其他的功能语句，可以在 ProU 用户手册中查看)

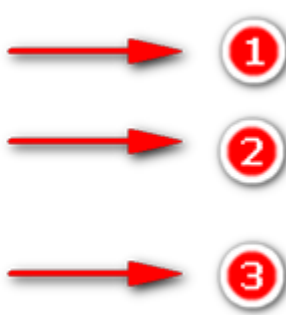


如果需要更改 IP 地址或者端口号, 可以依据本例子中的 IP 进行替换。



3、之后可在每段程序中，编写相应的程序，以实现打开网口、关闭网口、发送/接收数据的功能。

```
//打开网口
unsigned int Elopen(unsigned long long para1, unsigned long long para2)
{
    if(!GC_Read_SM(481))
    {
        return 0;
    }
    handle = GC_Eth_Open(ip1, port1);
    if(handle > 0)
    {
        M[2101] = 1; //网口打开完成标志位置1
        M[2104] = 0; //网口关闭完成标志位置0
    }
    return 0;
}
```



- 1) 判断是否有调用 C 程序功能，如果没有，则直接结束程序，不运行后面的程序。如果有，则正常运行。
- 2) 使用打开网口功能语句，并将结果赋值给句柄变量。如果打开成功，则句柄值大于 0，如果打开失败，则反之。
- 3) 如果网口打开成功，则置位打开完成标志位，复位关闭完成标志位。

```
//关闭网口
unsigned int Elclose(unsigned long long para1, unsigned long long para2)
{
    if(!GC_Read_SM(481))
    {
        return 0;
    }
    int n;
    n = GC_Eth_Close(handle);
    M[2101] = 0; //网口打开完成置0
    M[2104] = 1; //网口关闭完成置1
    return 0;
}
```

- 1) 执行关闭网口功能语句，执行后，复位网口打开完成标志位，置位网口关闭完成标志位。



```
//发送数据
unsigned int Elsend(unsigned long long para1, unsigned long long para2)
{
    if(!GC_Read_SM(481))
    {
        return 0;
        int idx;
        int n;
        for(idx = 0 ;idx < 14 ; idx++) //, 一个字存储2个字符x14=28字符
        {
            s1_Buff1[0 + idx * 2] = RD[0 + idx] ;
            s1_Buff1[1 + idx * 2] = (RD[0 + idx] >> 8) ;
        }
        n = GC_Eth_Send( handle, s1_Buff1, 29);
        if(n > 0)
        {
            M[2107] = 1;//发送正确
            M[2108] = 0;//发送错误
        }
        else
        {
            M[2107] = 0;
            M[2108] = 1;
        }
        return 0;
    }
}
```

- 1) 将 RD0~RD13 中的高低八位数据传送给发送缓存位，如果需要增大发送数据字符长度，需要更改此处的字长，还有全局变量和接收功能语句中的字符长。每个字元件对应两个字符位，所以程序开头的全局变量字符需要两倍于此处的字长。否则会出现数据丢失。
- 2) 发送数据，格式为句柄、缓存位、字符长度。并将发送数据的结果赋值给 n，如果发送成功，则 n>0，置位发送完成信号，复位发送错误信号；如果发送失败，则反之。



```
//接收数据
unsigned int Elrev(unsigned long long para1, unsigned long long para2)
{
    if(!GC_Read_SM(481))
    {
        return 0;
    }
    int idx;

    ① int n = GC_Eth_Rev(handle, r1_Buff1, 21); //设定接收字长

    ② for(idx = 0 ;idx < 10 ; idx++) //一个字存储2个字符x10=20字符
    {
        RD[200 + idx] = (r1_Buff1[0 + idx * 2] ) | (r1_Buff1[1 + idx * 2] <<8) ;
    }

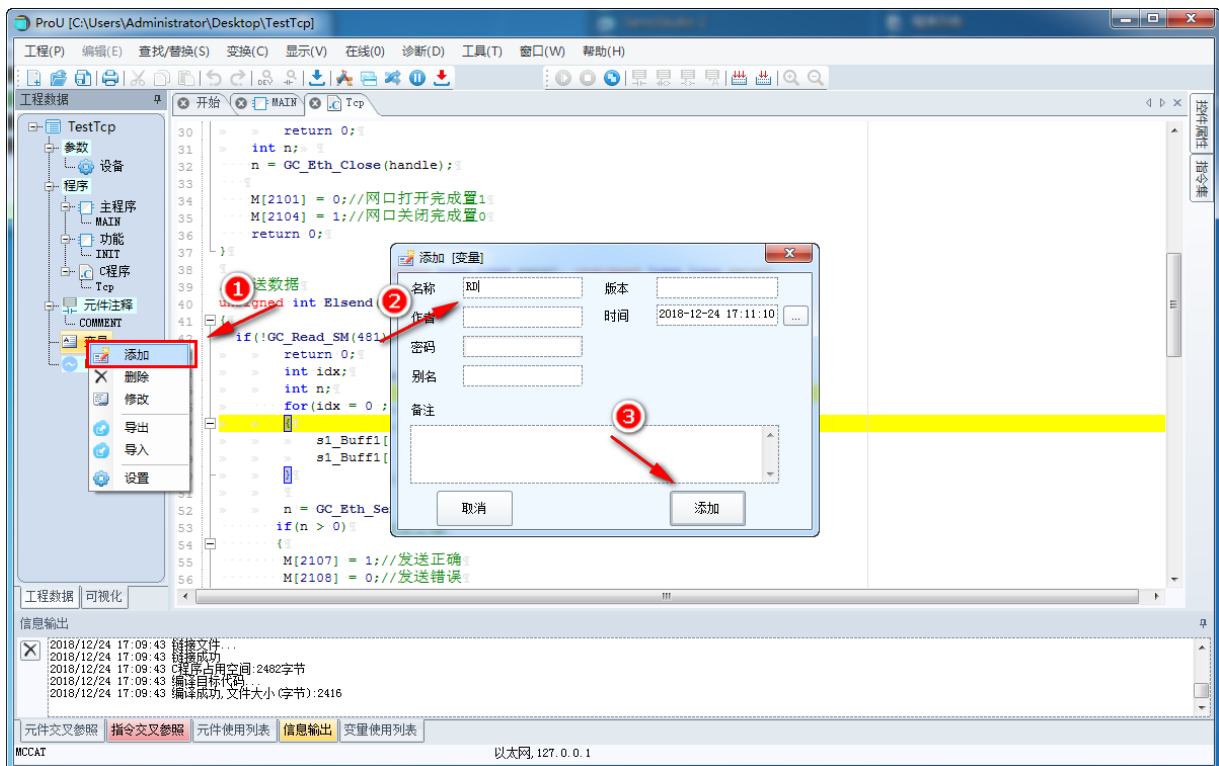
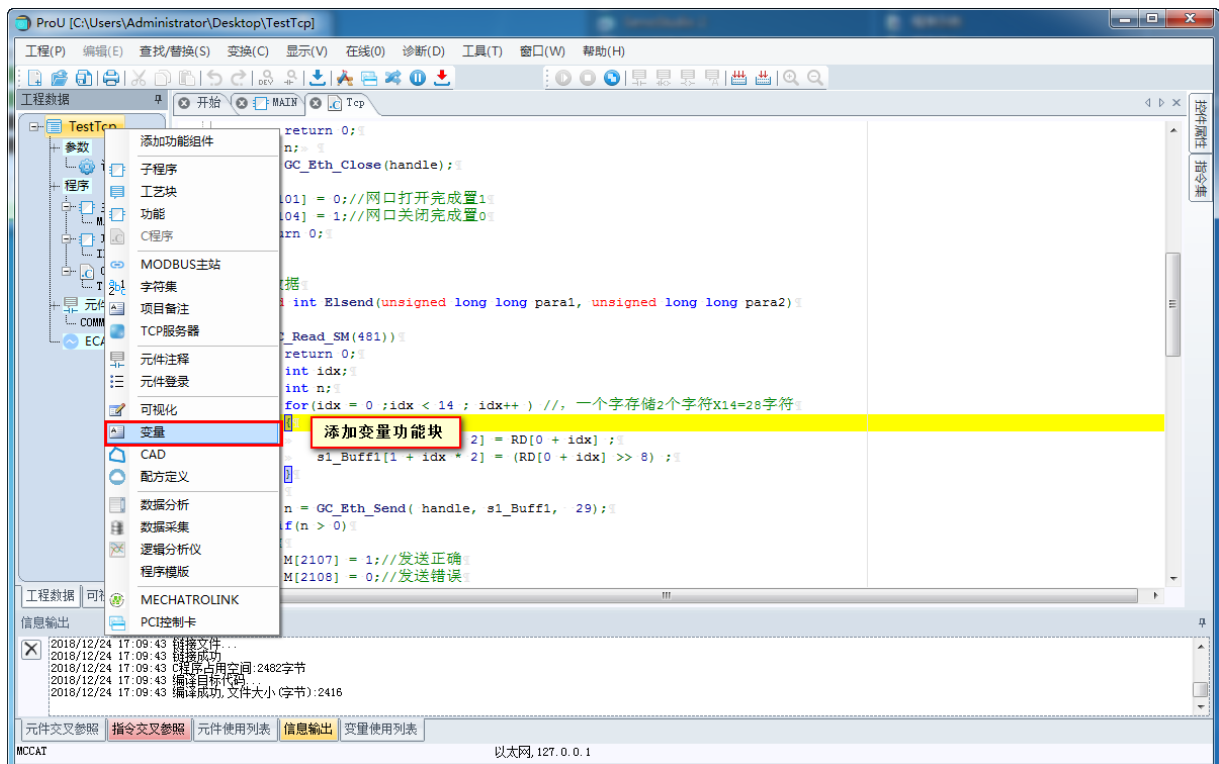
    if(n >0)
    {
        M[2110] = 1; //接收正确
        M[2111] = 0; //接收错误
    }
    ③ else
    {
        M[2110] = 0;
        M[2111] = 1;
    }
    return 0;
}
```

- 1) 接收数据，格式为句柄、接收缓存位、字符长度。并将数据存储在接收缓存位中、接收结果赋值给 n。
- 2) 将字符内的数据传送给字元件 RD200~RD209。与接收数据相同，如果想增加接收数据长度，则需要更改此处的字长、全局变量与接收功能语句的字符长。
- 3) 如果接收成功，则 n>0，并置位接收完成标志位，复位接收失败标志位。如果接收失败，则反之。



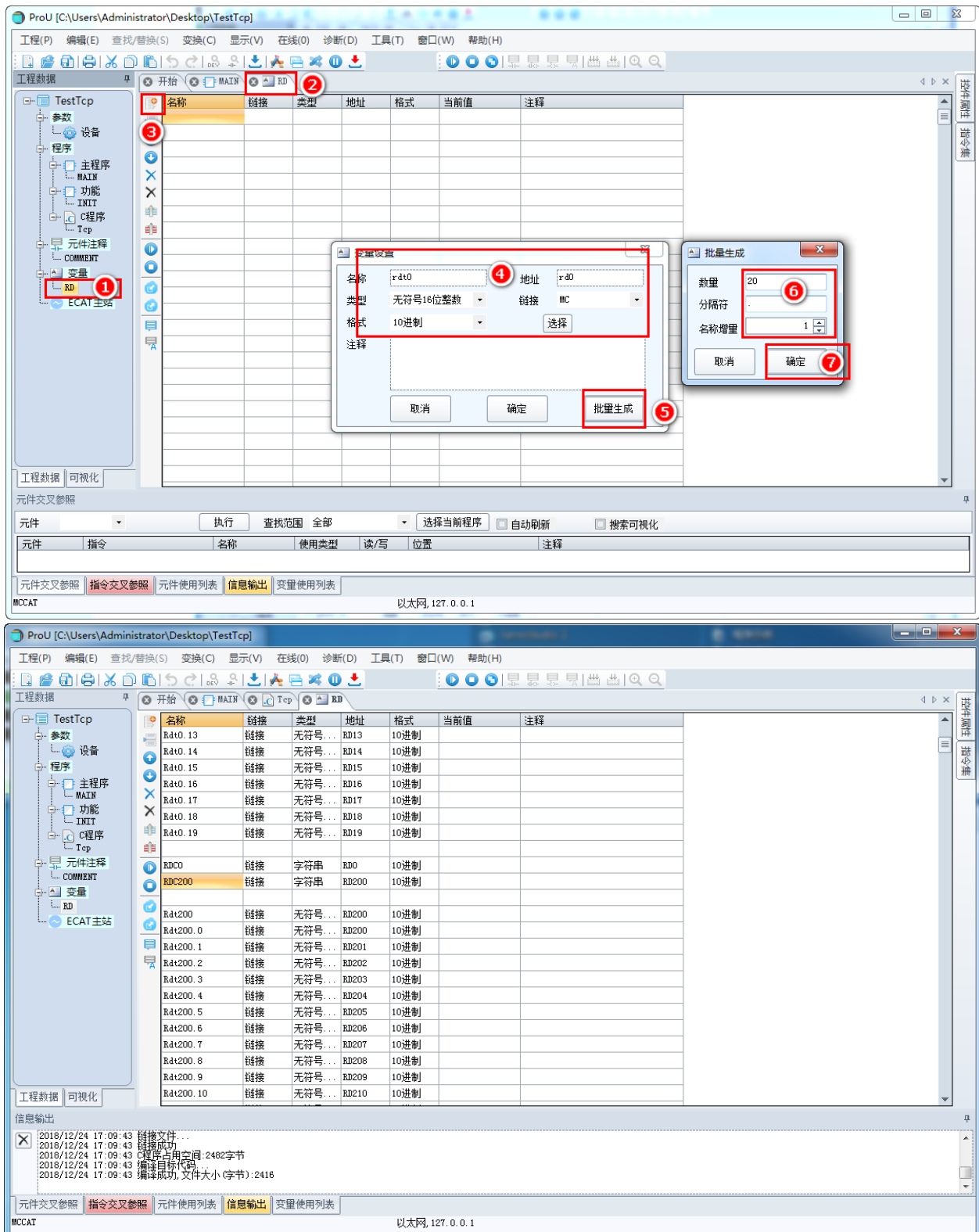
1.3 设置变量

1、在左侧项目数据中添加变量功能块，再添加变量。





- 2、C 语言中所使用的 RD 字元件设置变量，以便在可视化中调用。可以尽可能设置长一点，以便自由增减字符长度。如在本段 C 程序中使用了 RD0~RD13、RD200~RD209。

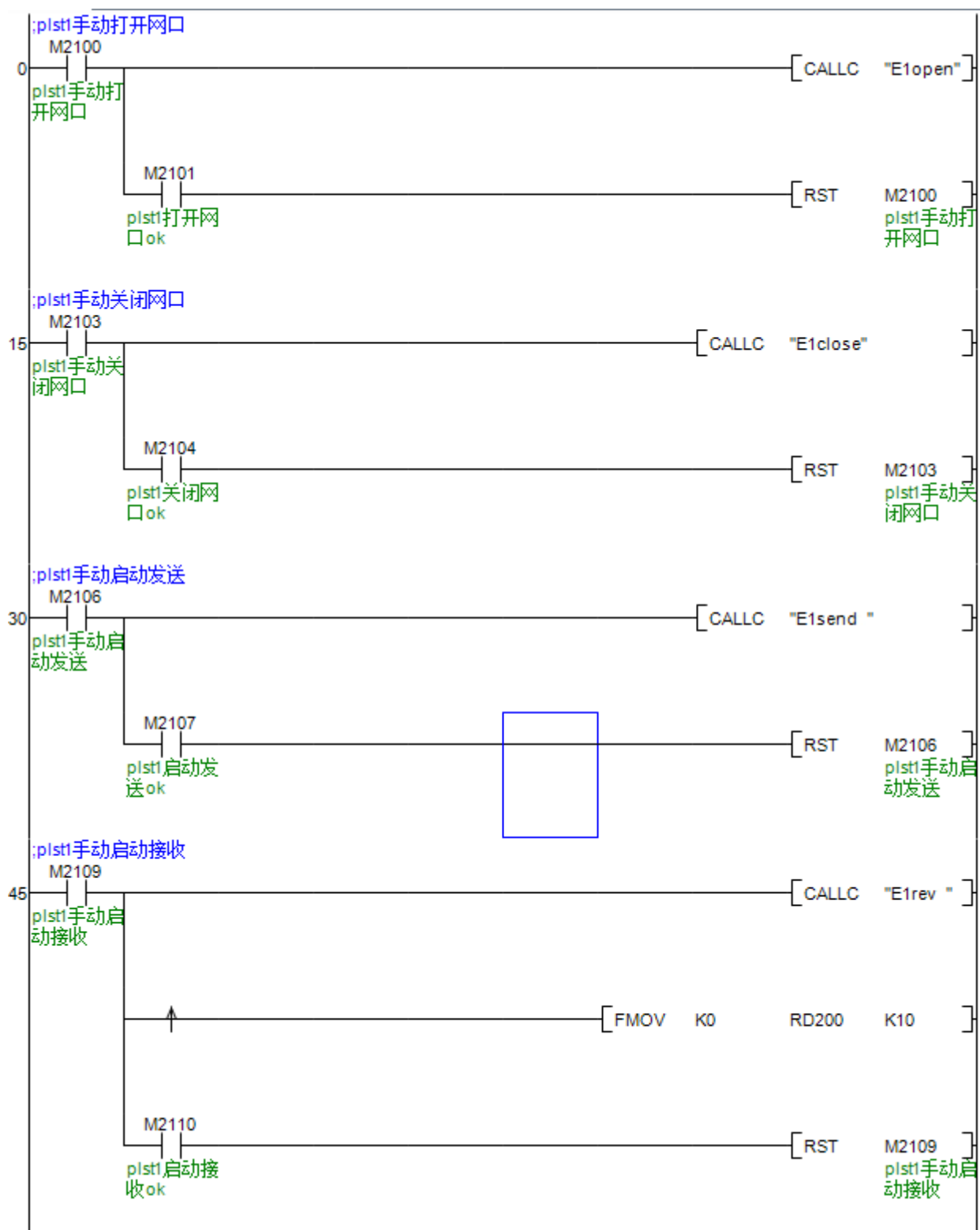


其中的 RDC0、RDC200 是 RD0、RD200 的字符串变量。



1.4 编写梯形图

1、编写梯形图，调用 C 程序

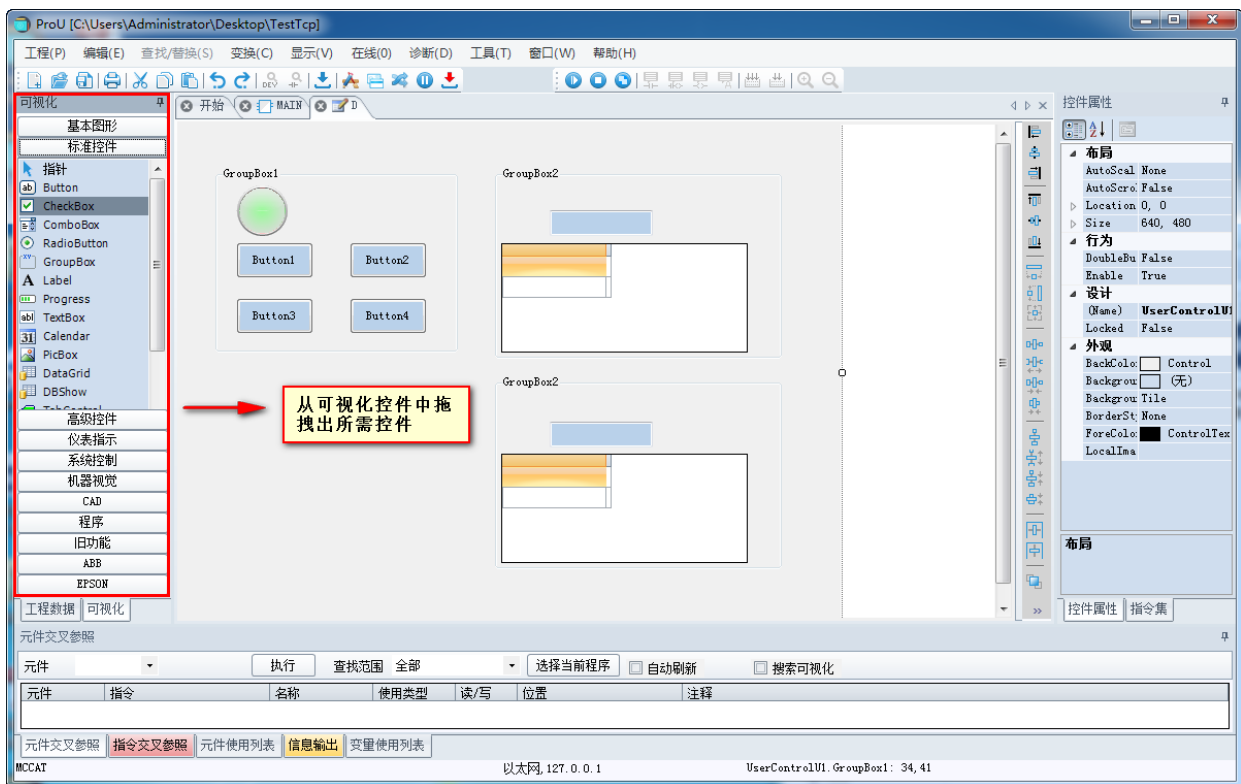
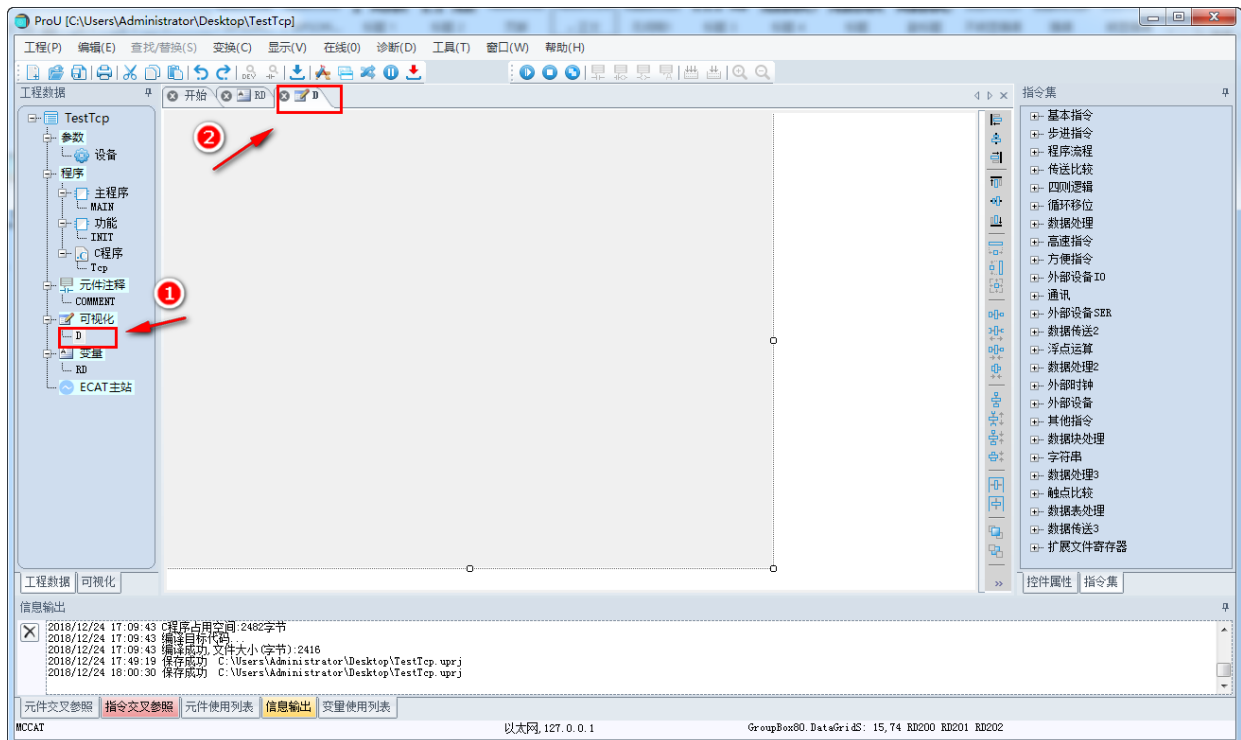




1.5 制作可视化界面

1、新建一个可视化界面。从可视化控件中拖出以下控件，然后填入相关参数。

1) Button: 4 个、Textstr: 2 个、Datagrid: 2 个、GroupBox: 3 个、DispBit(仪表指示中): 1 个





2) 控件参数:

Button:

布局	布局	布局	布局
Dock: None	Dock: None	Dock: None	Dock: None
Location: 15, 66	Location: 119, 66	Location: 15, 128	Location: 119, 128
Size: 95, 40	Size: 95, 40	Size: 95, 40	Size: 95, 40
设计	设计	设计	设计
(Name): Button22	(Name): Button23	(Name): Button17	(Name): Button20
Action: M2100	Action: M2103	Action: M2106	Action: M2109
Click: ON-OFF	Click: ON-OFF	Click: ON-OFF	Click: TOGGLE
Disable:	Disable:	Disable:	Disable:
Level: Level1	Level: Level1	Level: Level1	Level: Level1
Locked: False	Locked: False	Locked: False	Locked: False
LogEna: False	LogEna: False	LogEna: False	LogEna: False
LogMsg:	LogMsg:	LogMsg:	LogMsg:
PressTim: 0	PressTim: 0	PressTim: 0	PressTim: 0
Visible:	Visible:	Visible:	Visible:
外观	外观	外观	外观
BackColor: GradientA	BackColor: GradientA	BackColor: GradientA	BackColor: GradientA
BackColor: Silver	BackColor: Silver	BackColor: 255, 255, 255	BackColor: 255, 255, 255
BackColor: 128, 255, 255	BackColor: 128, 255, 255	BackColor: 128, 255, 255	BackColor: 128, 255, 255
BackColor: M2101	BackColor: M2104	BackColor: M2107	BackColor: M2110
Backgrou: (无)	Backgrou: (无)	Backgrou: (无)	Backgrou: (无)
Backgrou: Tile	Backgrou: Tile	Backgrou: Tile	Backgrou: Tile
Font: 宋体, 9pt	Font: 宋体, 9pt	Font: 宋体, 9pt	Font: 宋体, 9pt
ForeColor: ControlText	ForeColor: ControlText	ForeColor: ControlText	ForeColor: ControlText
ForeColor:	ForeColor:	ForeColor:	ForeColor:
ForeColor: Color[] Array	ForeColor: Color[] Array	ForeColor: Color[] Array	ForeColor: Color[] Array
LocalIma:	LocalIma:	LocalIma:	LocalIma:
LocalIma:	LocalIma:	LocalIma:	LocalIma:
Text: 打开网口	Text: 关闭网口	Text: 发送启动	Text: 接收启动
Text Off: 打开网口NG	Text Off: 关闭网口NG	Text Off: 发送数据	Text Off: 接收开启
Text On: 打开网口OK	Text On: 关闭网口OK	Text On: 发送启动OK	Text On: 接收数据OK
TextAlig: MiddleCenter	TextAlig: MiddleCenter	TextAlig: MiddleCenter	TextAlig: MiddleCenter

DispBit (指示灯) 参数:

Frame	
AddGlassEffect	True
ClipOuterFrame	True
FrameColor1	White
FrameColor2	DimGray
Style	Circular
Indicator	
BorderColor	Black
BorderWidth	0
Color1	255, 128, 128
Color2	Lime
FillType	None
ForeColor	Black
Style	Circular
Text	
布局	
Dock	None
Location	15, 20
Size	33, 35
设计	
(Name)	DispBit12
Locked	False
Read Var	M2101
外观	
BackColor	Transparent
Font	Microsoft Sans Serif, 12pt
Text Off	OFF
Text On	ON



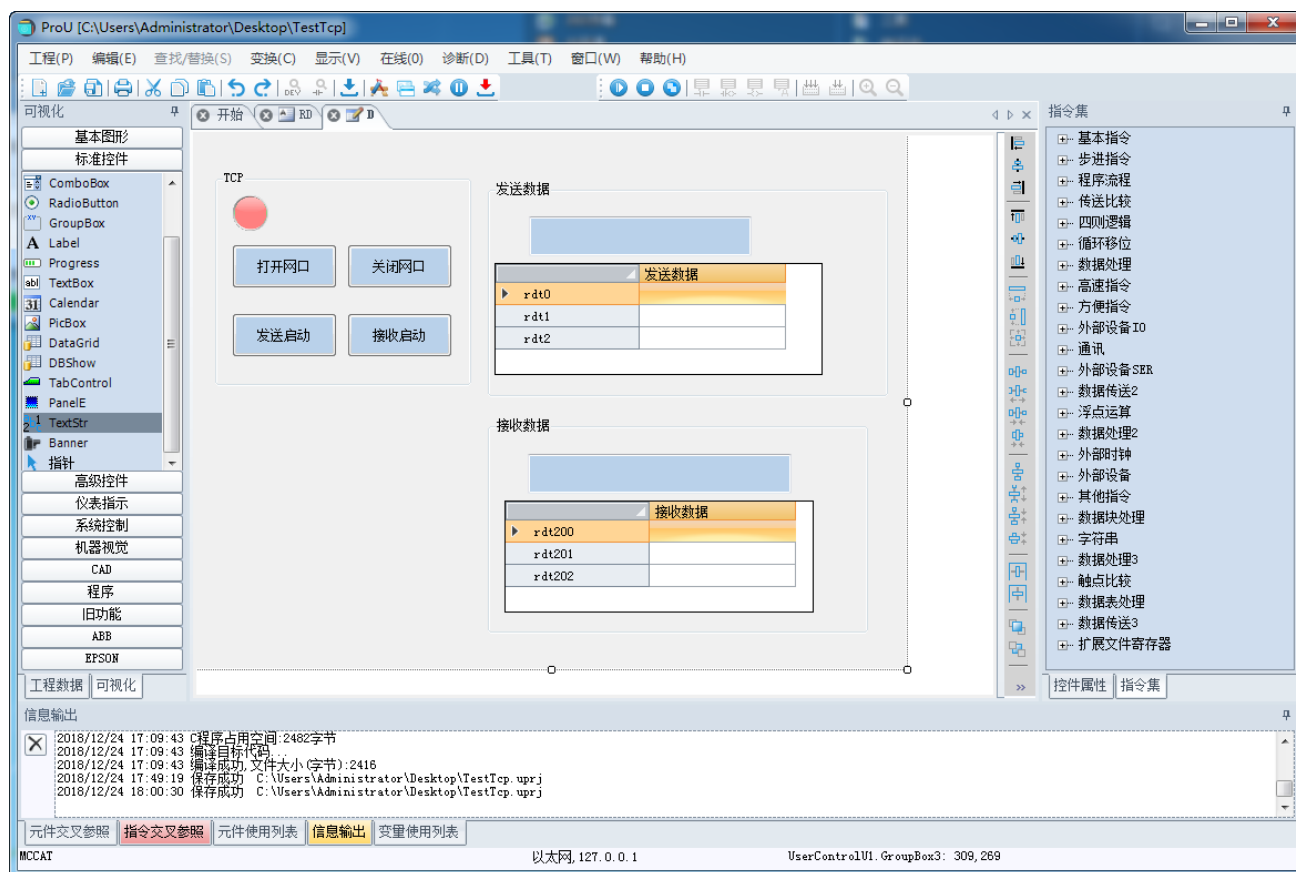
发送数据控件组参数:

布局		布局	
Dock	None	AutoScroll	False
Location	37, 31	Dock	None
Size	201, 36	Location	6, 73
行为		Size	296, 101
PasswordChar		行为	
ReadOnly	False	Enable	True
设计		设计	
(Name)	TextStr6	(Name)	DataGrid13
DataVar	RDC0	Cols	1
Length	32	ColText	发送数据
Locked	False	ColWidth	132
ReadString	False	DataDst	rdt0
Trig		DataSrc	rdt0
外观		DotNumS	
Align	Left	Level	Level2
BackColor	<input type="checkbox"/> GradientActiveCaption	Locked	False
BorderStyle	Fixed3D	RowHeadersWidth	130
Font	宋体, 9pt	Rows	3
ForeColor	<input checked="" type="checkbox"/> WindowText	RowText	rdt0;rdt1;rdt2;rdt3;
Text		TextS	String[] Array
需要更改的参数		外观	
		BackColor0	<input type="checkbox"/>
		BackColor1	<input type="checkbox"/>
		BackColor2~7	Color[] Array
		BorderStyle	None
		Font	宋体, 9pt
		ForeColor0	<input type="checkbox"/>
		ForeColor1	<input type="checkbox"/>
		ForeColor2~7	Color[] Array
		LocalImageOn	

接收数据控件组参数:

布局		布局	
Dock	None	AutoScroll	False
Location	36, 32	Dock	None
Size	238, 36	Location	15, 74
行为		Size	279, 101
PasswordChar		行为	
ReadOnly	False	Enable	True
设计		设计	
(Name)	TextStr5	(Name)	DataGrid12
DataVar	RDC200	Cols	1
Length	32	ColText	接收数据
Locked	False	ColWidth	132
ReadString	True	DataDst	rdt200
Trig		DataSrc	rdt200
外观		DotNumS	
Align	Left	Level	Level2
BackColor	<input type="checkbox"/> GradientActiveCaption	Locked	False
BorderStyle	Fixed3D	RowHeadersWidth	130
Font	宋体, 9pt	Rows	3
ForeColor	<input checked="" type="checkbox"/> WindowText	RowText	rdt200;rdt201;rdt202;rdt203
Text		TextS	String[] Array
需要更改的参数		外观	
		BackColor0	<input type="checkbox"/>
		BackColor1	<input type="checkbox"/>
		BackColor2~7	Color[] Array
		BorderStyle	None
		Font	宋体, 9pt
		ForeColor0	<input type="checkbox"/>
		ForeColor1	<input type="checkbox"/>
		ForeColor2~7	Color[] Array
		LocalImageOn	

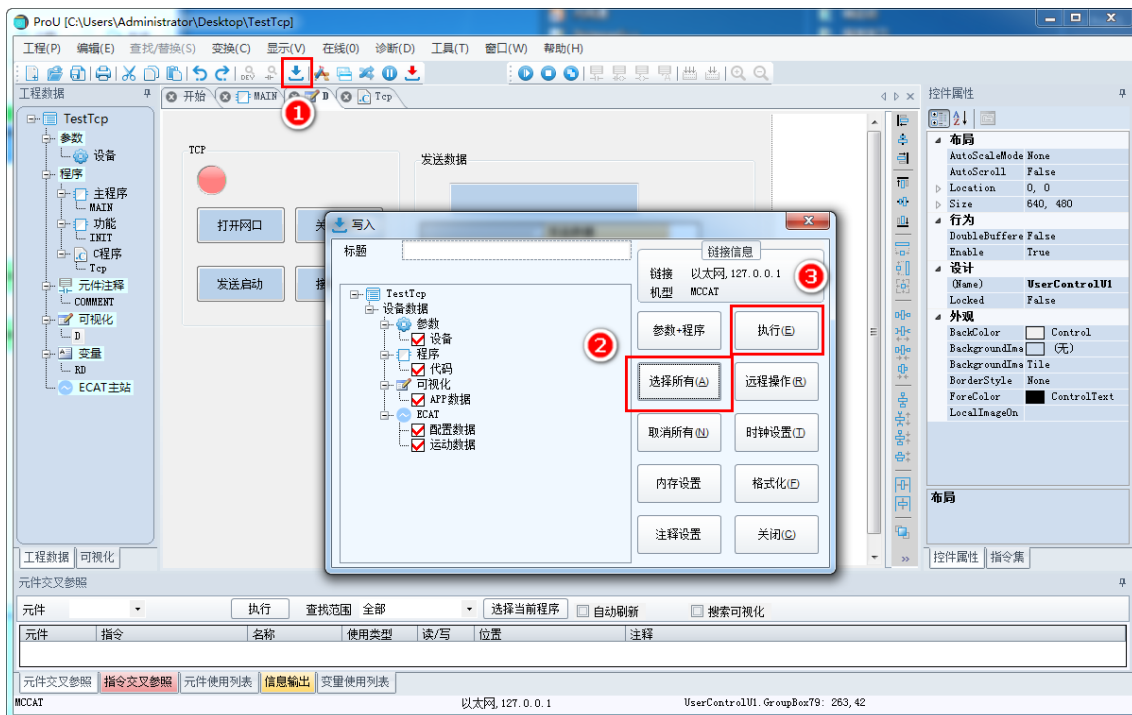
如果需要更改发送、接收的字符长度, 也需要修改 Length 的参数。





1.6 通讯测试

1、将编写好的 ProU 程序写入 ProH 中，写入完成后，重启 ProH，即会出现一个可供使用的可视化界面。





2、打开 SocketTool 工具



