

趋势图的历史显示

说明书

普洛菲斯国际贸易（上海）有限公司

技术热线：**021-6361-5008**

1.0 版

普洛菲斯国际贸易（上海）有限公司版权所有

目录

内容	页码
1. 概述.....	3
2. 画面操作.....	4
3. 画面建立.....	5
4. 地址和D脚本说明	7
5. GP、PLC和画面编辑软件版本.....	18
6. 画面复制.....	18
7. 注意事项.....	20
<附录>.....	21
建立和编辑D脚本.....	21
修改D脚本地址	23
关于LS区	25
确认地址	26
关于图形显示.....	27
关于CSV 数据传输.....	29
关于T-tag.....	31

注意：在您的系统中使用本例时，在操作前请检查。

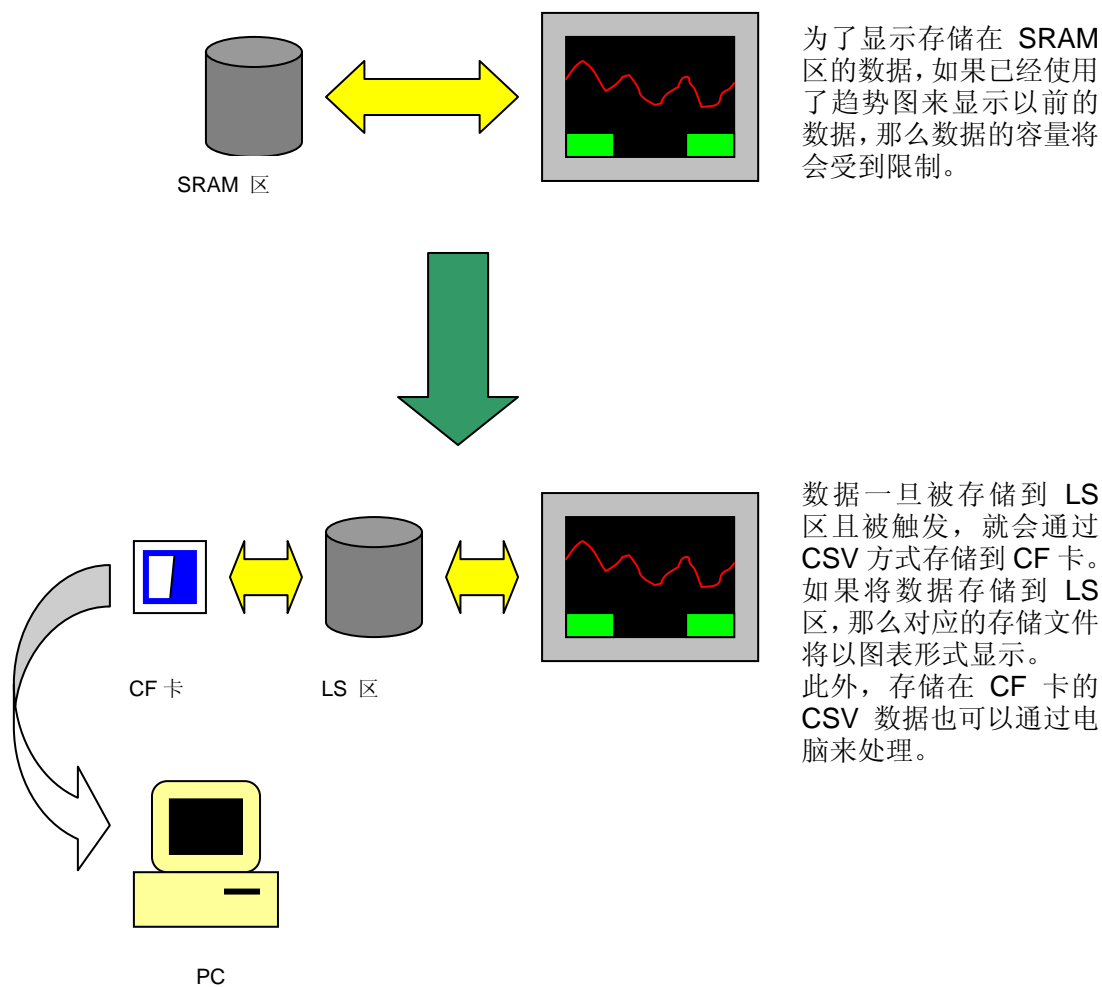
1. 概述

从 6.1 版开始增加了历史数据显示功能。如果需要，可以通过趋势图的方式显示以前的数据。但是，由于此时的存储位置是 GP 内部的 SRAM 区，因此受存储容量的限制。

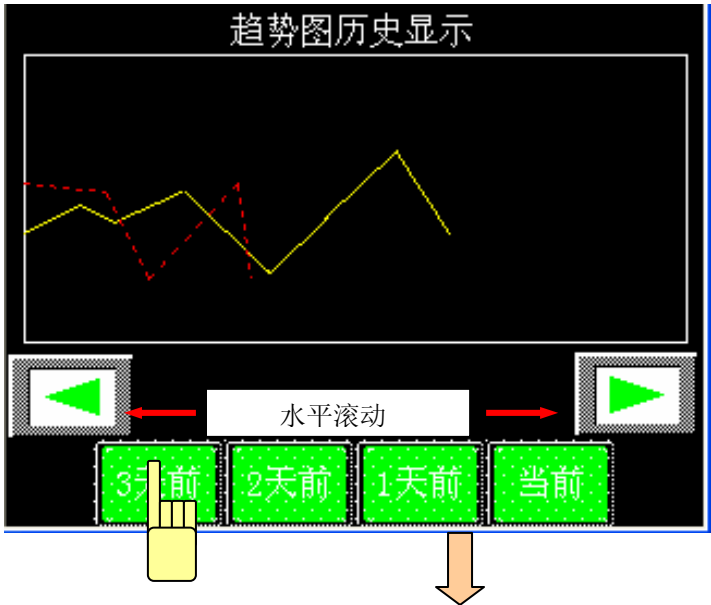
本例中使用了一项新功能，即 CSV 数据传输功能，它是从 6.1 版开始新增的一项功能。通过 CSV 方式可以将当前的数据存储到 CF 卡。如果需要，可以将数据调出来并以图形的方式重新显示。

用户可以使用 CF 卡来存储大量的数据。

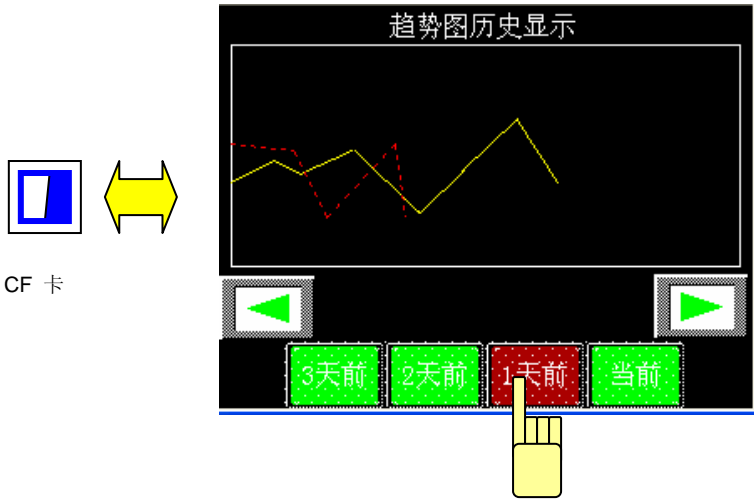
此外，可以通过电脑来编辑存储在 CF 卡中的数据。



2. 画面操作



如果按下“当前”按钮，将执行 1 分钟周期的图形显示。
如果显示的画面到达极限，图形将滚动，并从起始处显示。
当按下两边带有箭头的按钮时，将重新显示刚才显示的图形。
过去一天收集的数据将存储到 CF 卡。

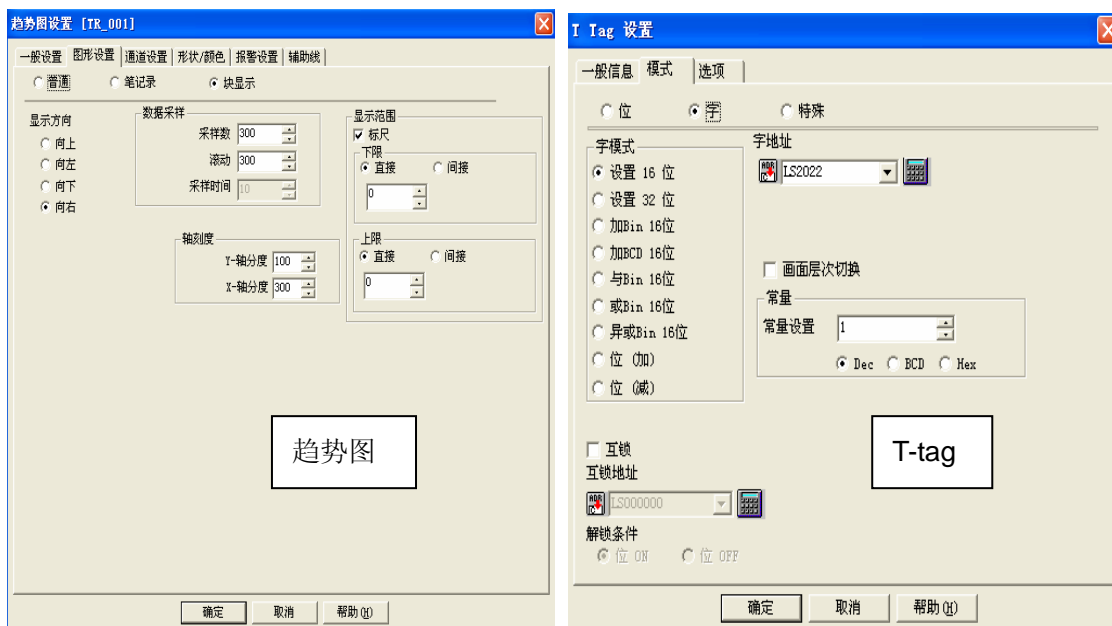
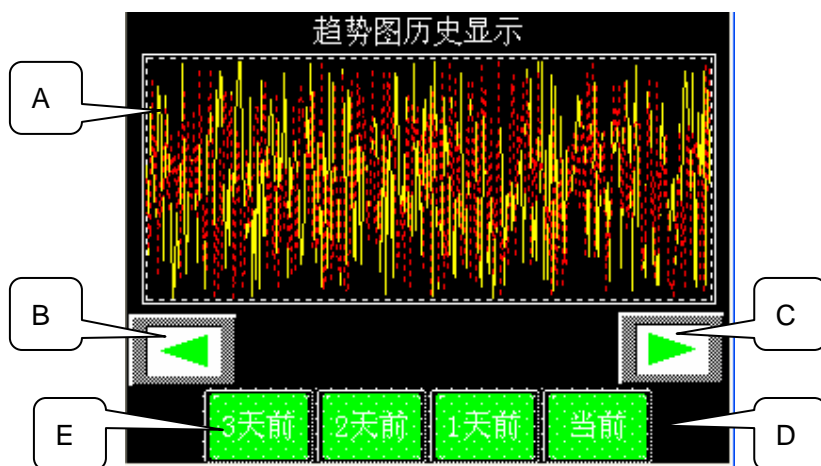


按下“一天前”按钮，数据将从 CF 卡中读出并以图表方式显示。
当按下两边带有箭头的按钮时，将重新显示刚才显示的图形。

3. 画面建立

B8999: 当前两边极限数据的设置。

- A: 趋势图 (块显示) 将显示存储在 GP 内部 LS 区的数据。
(→关于图形显示)
- B: 历史数据显示按钮 (向后) 读的区域将改变且重新显示图形。
- C: 历史数据显示按钮 (向前) 同上。
- D: 当前数据显示按钮 (T-tag) 当按下此按钮时, 将要求读数据。然后将读出存储在 LS 区的数据, 且以图形形式显示。(→关于 T-tag)
- E: CF 卡数据显示按钮 (T-tag) 按下此按钮将读取 CF 卡中的数据并存储到 LS 区, 然后以图形形式显示。



如果本例中的画面显示后，数据将以一分钟的周期存储到 LS 区，且趋势图将以块显示的方式显示出来。尽管趋势图的数据点是 300 点，但 300 点的数据显示完成后，将清空显示区域，然后再从 0 开始显示。此时，如果按下历史数据显示按钮，那么将显示刚才显示过的趋势图。在采集一天后（获得 1440 个数据），通过 CSV 数据传输方式将采集到的数据存储到 CF 卡。

（→关于 CSV 数据传输）

如果按下“一天前”按钮，数据将从 CF 卡读出，然后通过 CSV 数据传输方式写入 LS 区，并再次以图形形式显示。

如果按下“当前”按钮，将显示最新的数据。

注意：尽管在读和写 CF 卡时，趋势图将移动显示当前的数据，但是仍能正常显示历史数据。

如果在 GP 启动后立即读取 CF 卡中不存在的数据，将会发生错误。

如果电源被复位，LS 区存储的数据和文件编号将被清除。

您可以按照本例来建立 B8999（基本画面 8999），但请根据您的系统修改相应的画面编号。

（→6. 画面复制）

4. 地址和 D 脚本说明

本例中使用了 D 脚本。其说明如下：(→如何建立/编辑 D 脚本)

使用的地址位于 LS 区，如下所示：(→关于 LS 区) 请根据您的系统修改相应地址。(→如何修改 D 脚本地址)

使用的地址	详细说明
字地址	
LS1000	趋势图 CH1 的控制字
LS1001	趋势图 CH1 显示数据的数目
LS1002~1301	趋势图 CH1 的数据
LS1400	趋势图 CH2 的控制字
LS1401	趋势图 CH2 显示数据的数目
LS1402~1701	趋势图 CH2 的数据
LS2023	CSV 传输控制字
LS2024	CSV 传输状态
LS2025	CSV 传输文件编号
LS2026	CSV 传输模式
LS2027	CSV 传输设备代码
LS2028	CSV 传输设备地址
LS2029	CSV 传输数据数目
LS2430~3869	CH1_CF 数据显示的数据存储区
LS3870~5309	CH2_CF 数据显示的数据存储区
LS5310~6749	CH1 当前显示的数据存储区
LS6750~8189	CH2 当前显示的数据存储区
LS8190	CH1 虚拟数据
LS8191	CH2 虚拟数据
位地址	
LS202200	当前指示灯
LS202201	1 天前指示灯
LS202202	2 天前指示灯
LS202203	3 天前指示灯
LS240800	历史显示的向后按钮互锁
LS240801	历史显示的向前按钮互锁

临时地址	
t0076	在当前显示中累加 300 个数据的数目
t0078	历史显示的临时存储地址
t0079	为 CF 卡写错误处理循环
t0080	为 CF 卡读错误处理循环
t0081	CF 文件编号和新近写入编号
t0082	存入数据读出来的偏移量
t0083	当前数据存入的偏移量
t0084	为 0
t0085	当前显示的显示标记地址
t0086	历史显示的显示标记地址
t0087	当前显示读出数据的偏移量
t0088	CH2 虚拟数据的临时地址
t0089	CH1 虚拟数据的临时地址

D脚本程序说明

ID00000 存储数据建立

触发：每隔 60 秒，将执行下面的程序。

```

// CH1 虚拟数据建立。
if ( [w:LS8190] < 100 and [t:0089] == 0 ) //增加 0~100
{
    [w:LS8190] = [w:LS8190] + 1
    if ( [w:LS8190] == 100 )
    {
        [t:0089] = 1
    }endif
}
else
{
    [w:LS8190] = [w:LS8190] - 1
    if ( [w:LS8190] == 0 )
    {
        [t:0089] = 0
    }endif
}endif

// CH2 虚拟数据建立。
if ( [w:LS8191] < 300 and [t:0088] == 0 ) // 增加 0~300
{
    [w:LS8191] = [w:LS8191] + 2
    if ( [w:LS8191] == 300 )
    {
        [t:0088] = 1
    }endif
}
else
{
    [w:LS8191] = [w:LS8191] - 2
    if ( [w:LS8191] == 0 )
    {
        [t:0088] = 0
    }endif
}endif

//如果指定了 PLC 的值，它将取代 LS8190 和 LS8191。
_memcpy_EX([w:LS5310]#[t:0083], [w:LS8190]#[t:0084], 1)// CH1 数据存到 LS 区。
.
_memcpy_EX([w:LS6750]#[t:0083], [w:LS8191]#[t:0084], 1)//CH2 数据存到 LS 区。

//偏移量推进到 0-1439（总计 1440 个字）。
if ( [t:0083] < 1439 ) // 当存储区域的偏移量小于 1439 时执行。
{
    [t:0083] = [t:0083] + 1 //加 1
}
else //当存储区域的偏移量为 1440 时执行。
{
    [t:0083] = 0 //清 0
}endif

```

ID00001 趋势图的更新

```

触发：每隔 60 秒，将执行下面的程序。
if ( [w:LS2409] == 0 and [b:LS202200] == 1)           // 在当前数据显示期间执行。
{
  _memcpy_EX([w:LS1002]#[t:0085], [w:LS5310]#[t:0087], 1) // 从 CH1 的数据存储区复
                                                         // 制到图形绘制区。

  _memcpy_EX([w:LS1402]#[t:0085], [w:LS6750]#[t:0087], 1) // 从 CH2 的数据存储区复
                                                         // 制到图形绘制区。

  [w:LS1001] = [t:0085]+1           // CH1 图形绘制标记增加。
  [w:LS1401] = [t:0085]+1           // CH2 图形绘制标记增加。
  [w:LS1000] = 3                     //CH1 图形重画。
  [w:LS1400] = 1                     //CH2 图形重画。
}endif

if ( [t:0087] < 1440 )           //当存储区域的偏移量小于 1440 时执行。
{
  [t:0087] = [t:0087] + 1         //数据存储区域的偏移量移动。
  if ( [t:0087] == 1440 )         //当偏移值设定为 1440 时执行。
  {
    Call CF_Write                 //函数调用。
    [t:0079] = 10000              // 设定重复的次数。
    loop([t:0079])                //重复下面的程序。
    {
      if ( [b:LS202401] == 1 )    //当 CSV 传输状态变成传输完成时，将执行下面
                                  //的程序。
      {
        [w:LS2023] = 0           // 清除 CSV 传输触发。
        break                   // 退出循环。
      }endif
    }endloop
    [t:0087] = 0                 // 数据存储的偏移值初始化。
    [t:0076] = 0                 //初始化当前数据（300）收集的次数。
    [t:0085] = 0                 //初始化图形显示的偏移值。
  }endif
}endif

if ( [t:0085] < 300 )           //当图形显示的偏移值小于 300 时执行。
{
  [t:0085] = [t:0085] + 1       //图形显示的偏移量移动。
}
else                             //当图形显示的偏移值设定为 300 时执行。
{
  [t:0076] = [t:0076] + 1       // 当存储 300 当前数据后加 1。
  [t:0085] = 0                 //初始化图形显示的偏移值。
}endif

```

ID00002 历史显示

```

触发: ([w: LS2409]<> [t:0078]) 当历史按钮的数值不等于临时存储的数值时执行。
if ( [w:LS2022] == 1 ) // 当显示当前数据时执行。

{
  if ( [t:0087] > 300 ) //当当前数据的存储地址的偏移量大于 300 时执行。

  {
    if ( [w:LS2409] == 0 ) // 当没有触摸历史显示按钮时执行。

    {
      Call present_data // 函数调用。
    }
    else //当触摸历史显示按钮时执行。
    {
      [t:0082] = ( [t:0076] - [w:LS2409] ) * 300 // 存储 300 当前数据的次数和历史显示按钮数值的差值乘 300。

// CH1 当前存入数据的读出来的首地址改变, 且它取代了图形显示地址。
. _memcpy_EX([w:LS1002]#[t:0086],[w:LS5310]#[t:0082],300)
// CH2 当前存入数据的读出来的首地址改变, 且它取代了图形显示地址。
_memcpy_EX([w:LS1402]#[t:0086],[w:LS6750]#[t:0082],300)
      [w:LS1001] = 300 // CH1 图形显示标记。
      [w:LS1401] = 300 // CH2 图形显示标记。
      [w:LS1000] = 3 // CH1 图形重新绘制。
      [w:LS1400] = 1 // CH2 图形绘制。
    }
  }
}endif
}endif
}
else // 当显示 CF 卡数据时执行。
{
  if ( [w:LS2022] >= 2 ) //当显示 CF 卡数据时执行。
  {
    if ( [w:LS2409] == 0 ) //没有按下历史显示按钮时执行。
    {
      [t:0082] = 1200 //CF 卡数据读出来的首地址的偏移值。
// 从 CH1_CF 存储数据读出来的首地址加 1200 开始的 300 个字取代图形显示地址。
_memcpy_EX([w:LS1002]#[t:0086], [w:LS2430]#[t:0082], 240)
//从 CH2_CF 存储数据读出来的首地址加 1200 开始的 300 个字取代图形显示地址。

_memcpy_EX([w:LS1402]#[t:0086], [w:LS3870]#[t:0082], 240)
      [w:LS1001] = 240 // CH1 图形显示标记。
      [w:LS1401] = 240 // CH2 图形显示标记。
      [w:LS1000] = 3 // CH1 图形重新绘制。
      [w:LS1400] = 1 // CH2 图表绘制。
    }
  }
}
else //当按下历史显示按钮时执行。
{
  [t:0082] = ( 4 - [w:LS2409] ) * 300 //4 和历史显示按钮的差值乘 300。
}
}

```

// CH1 CF 卡存储数据读出来的首地址被改变，且取代一个图形显示地址。

```

    _memcpy_EX([w:LS1002]#[t:0086], [w:LS2430]#[t:0082], 300)
// CH2 CF 存储数据读出来的首地址被改变，且取代一个图形显示地址。

```

```

    _memcpy_EX([w:LS1402]#[t:0086], [w:LS3870]#[t:0082], 300)

```

```

    [w:LS1001] = 300           //CH1 图形显示标记。
    [w:LS1401] = 300         //CH2 图形显示标记。
    [w:LS1000] = 3           //CH1 图形重新绘制。
    [w:LS1400] = 1           //CH2 图表绘制。
}endif
}endif
}endif
[t:0078] = [w:LS2409]       // 历史显示按钮的数值被临时存储。

```

ID00003 互锁

触发: ([b:LS203200]) 只要有一个通讯周期，就会执行上升沿和下降沿。

```

if ( [w:LS2409] == 0 )      //没有按下历史显示按钮时执行。
{
    set([b:LS240801])       //互锁开。
}
else                        //当按下历史显示按钮时执行。
{
    clear([b:LS240801])    // 互锁关。
}endif

if ( [w:LS2022] == 1 )     // 当显示当前数据时执行。
{
    if ( [w:LS2409] == [t:0076] or [t:0087] <= 300 ) //按历史显示按钮的次数是否和 300 当前数
                                                    //据存储次数相等?

    {
        set([b:LS240800])   // 互锁开。
    }
    else                    //按历史显示按钮的次数和 300 当前数
                            //据存储次数不相等。

    {
        clear([b:LS240800]) // 互锁关。
    }endif
}
else                        //当显示 CF 卡数据时执行。
{
    if ( [w:LS2409] == 4 )  // 当历史显示按钮的值设定为 4 时执行。

    {
        set([b:LS240800])   // 互锁开。
    }
    else
    {

```

```

clear([b:LS240800])           // 互锁关。
}endif
}endif

```

ID00004 1 天前

```

触发: ([b: LS202201])上升沿           当按下一天前按钮时执行。
if ( [t:0081] == 0 )                 //当新建的文件编号为 0 时执行。
{
[w:LS2025] = 0                       //它被看成文件编号为 0。
}
else                                  //当新建的文件编号非 0 时执行。
{
if ( [t:0081] == 1 )                 //当新建的文件编号为 1 时执行。
{
[w:LS2025] = 1                       //它被看成文件编号为 1。
}
else                                  //当新建的文件编号为 2 时执行。
{
[w:LS2025] = 2                       //它被看成文件编号为 2。
}endif
}endif
Call CF_Read                         //函数调用。

```

ID00005 2 天前

```

触发: ([b: LS202202]) 上升沿         当按下 2 天前按钮时执行。
if ( [t:0081] == 0 )                 //当新建的文件编号为 0 时执行。
{
[w:LS2025] = 2                       //它被看成文件编号为 2。
}
else                                  //当新建的文件编号非 0 时执行。
{
if ( [t:0081] == 1 )                 //当新建的文件编号为 1 时执行。
{
[w:LS2025] = 0                       //它被看成文件编号为 0。
}
else                                  //当新建的文件编号为 2 时执行。
{
[w:LS2025] = 1                       //它被看成文件编号为 1 时执行。
}endif
}endif
Call CF_Read                         //函数调用。

```

ID00006 3 天前

```

触发: ([b: LS202203])上升沿           当按下 3 天前按钮时执行。
if ( [t:0081] == 0 )                 // 当新建的文件编号为 0 时执行。
{
[w:LS2025] = 1                       //它被看成文件编号为 1。
}
else                                  //当新建的文件编号非 0 时执行。
{
if ( [t:0081] == 1 )                 //当新建的文件编号为 1 时执行。
{

```

```

[w:LS2025] = 2           //它被看成文件编号为 2。
}
else                     //当新建的文件编号为 2 时执行。
{
  [w:LS2025] = 0         //它被看成文件编号为 0。
}endif
}endif
Call CF_Read             //函数调用。

```

ID00007 当前显示

触发: ([b:LS202200])上升沿 当变成当前显示时, 就会执行以下程序。

```

if ( [t:0087] >= 1200) //当数据存储的偏移值为 1200 或更大时执行。
{
  memcpy([w:LS1002],[w:LS6510],300) // CH1 的数据存储地址的 300 个字取代图形显示区域。

  memcpy([w:LS1402],[w:LS7950],300) // CH2 的数据存储地址的 300 个字取代图形显示区域。
}
else //当数据存储的偏移值小于 1200 时执行。
{
  if ( [t:0087] >= 900) //当数据存储的偏移值为 900 或更大时执行。
  {
    // CH1 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1002],[w:LS6210],300) // CH2 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1402],[w:LS7650],300)
  }
else //当数据存储的偏移值小于 900 时执行。
{
  if ( [t:0087] >= 600) //当数据存储的偏移值为 600 或更大时执行。
  {
    // CH1 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1002],[w:LS5910],300) // CH2 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1402],[w:LS7350],300)
  }
else //当数据存储的偏移值小于 600 时执行。
{
  if ( [t:0087] >= 300) //当数据存储的偏移值为 300 或更大时执行。
  {
    //CH1 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1002],[w:LS5610],300) // CH2 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1402],[w:LS7050],300)
  }
}
}
}
}

```

```

else      //当数据存储的偏移值小于 300 时执行。
{
    // CH1 的数据存储地址的 300 个字取代图形显示
    // 区域。
    memcpy([w:LS1002],[w:LS5310],300)
    // CH2 的数据存储地址的 300 个字取代图形显示
    // 区域。
    memcpy([w:LS1402],[w:LS6750],300)
}endif
}endif
}endif
}endif

// 趋势图数据的数目。
[w:LS1001] = [t:0085] // CH1 图形显示标记。
[w:LS1401] = [t:0085] // CH2 图形显示标记。
// 显示控制。
[w:LS1000] = 3 // CH1 图形重新绘制。
[w:LS1400] = 1 // CH2 图表重新绘制。

```

ID00008 画面开始

触发: ([w:LS0000]= =8999) 如画面设置是 8999, 它将执行一次。

```

[w:LS2022]=1 // 当前显示。
[w:LS2409]=0 // 清除历史显示。

```

函数 CF_Read

```

[w:LS2026] = 0 // CSV 传输模式的格式。
[w:LS2027] = 0x4000 // CSV 传输设备代码。
[w:LS2028] = 2430 // CSV 传输地址。
[w:LS2029] = 2880 //CSV 传输数据的数目。
[w:LS2023] = 1 // CSV 传输控制。
[t:0080] = 10000 // 设定重复次数。
loop([t:0080]) //下面的程序将重复执行。
{
    if ( [b:LS202401] == 1 ) // 当 CSV 传输状态变成传输完成时执行。
    {
        [w:LS2023] = 0 // CSV 传输控制被清 0。
        break //退出循环。
    }endif
}endloop

// CF 卡数据中的 240 个字取代图形显示区域存储到 CH1
// 的 LS 区。
memcpy([w:LS1002],[w:LS3630],240)
// CF 卡数据中的 240 个字取代图形显示区域存储到 CH2
// 的 LS 区。
memcpy([w:LS1402],[w:LS5070],240)
[w:LS1001] = 240 // CH1 图形显示标记。
[w:LS1401] = 240 // CH2 图形显示标记。
[w:LS1000] = 3 // CH1 图形重新绘制。
[w:LS1400] = 1 // CH2 图形重新绘制。
[w:LS2409] = 0 //历史显示按钮的值被清 0。
[w:LS2025] = [t:0081] // 指定 CSV 传输文件编号。

```

函数 CF_Write

```

[w:LS2026] = 0           //CSV 传输模式格式。
[w:LS2027] = 0x4000     //CSV 传输设备代码。
[w:LS2028] = 5310      // 执行 CSV 传输的首地址。
[w:LS2029] = 2880      //CSV 传输数据的数目。
[w:LS2023] = 2         //CSV 传输控制。
if ( [t:0081] < 2 )    // 当文件编号小于 2 时执行。
{
    [t:0081] =[t:0081] + 1    //文件编号加 1。
}
else                    //当文件编号为 2 时执行。
{
    [t:0081] = 0             //文件编号设为 0。
}endif
[w:LS2025] = [t:0081] // CF 卡传输时的文件编号。

```

函数 present_data

```

if ( [t:0087] >= 1200) //当数据存储的偏移值为 1200 或更大时执行。
{
    memcpy([w:LS1002],[w:LS6510],300) //CH1 的数据存储地址的 300 个字取代图形显示区域。
    memcpy([w:LS1402],[w:LS7950],300) // CH2 的数据存储地址的 300 个字取代图形显示区域。
}
else //当数据存储的偏移值小于 1200 时执行。
{
    if ( [t:0087] >= 900) //当数据存储的偏移值为 900 或更大时执行。
    {
        // CH1 的数据存储地址的 300 个字取代图形显示区域。
        memcpy([w:LS1002],[w:LS6210],300)
        // CH2 的数据存储地址的 300 个字取代图形显示区域。
        memcpy([w:LS1402],[w:LS7650],300)
    }
    else //当数据存储的偏移值小于 900 时执行。
    {
        if ( [t:0087] >= 600) //当数据存储的偏移值为 600 或更大时执行。
        {
            //CH1 的数据存储地址的 300 个字取代图形显示区域。
            memcpy([w:LS1002],[w:LS5910],300)
            // CH2 的数据存储地址的 300 个字取代图形显示区域。
            memcpy([w:LS1402],[w:LS7350],300)
        }
        else //当数据存储的偏移值小于 600 时执行。
        {
            if ( [t:0087] >= 300) // 当数据存储的偏移值为 300 或更大时执行。

```



```
{  
// CH1 的数据存储地址的 300 个字取代图形显示区域。  
    memcpy([w:LS1002],[w:LS5610],300)  
// CH2 的数据存储地址的 300 个字取代图形显示区域。  
    memcpy([w:LS1402],[w:LS7050],300)  
}endif  
}endif  
}endif  
[w:LS1001] = [t:0085]           //CH1 图形显示标记。  
[w:LS1401] = [t:0085]           //CH2 图形显示标记。  
[w:LS1000] = 3                   // CH1 图形重新绘制。  
[w:LS1400] = 1                   // CH2 图形绘制。
```

5. GP、PLC 和画面编辑软件版本

本例中使用的 GP、PLC 和画面编辑软件版本的说明如下。请根据用户使用的型号，适当地调整画面的位置和相关设置。

(→7. 注意事项)

使用的GP和PLC

GP: GP2301S

PLC: MELSEC A1SJH (通讯模块), 三菱电机

(协议: MITSUBISHI MELSEC-AnN(LINK))

画面编辑软件版本: GP-PRO/PBIII C-Package03 (V7.23 中文版)

6. 画面复制

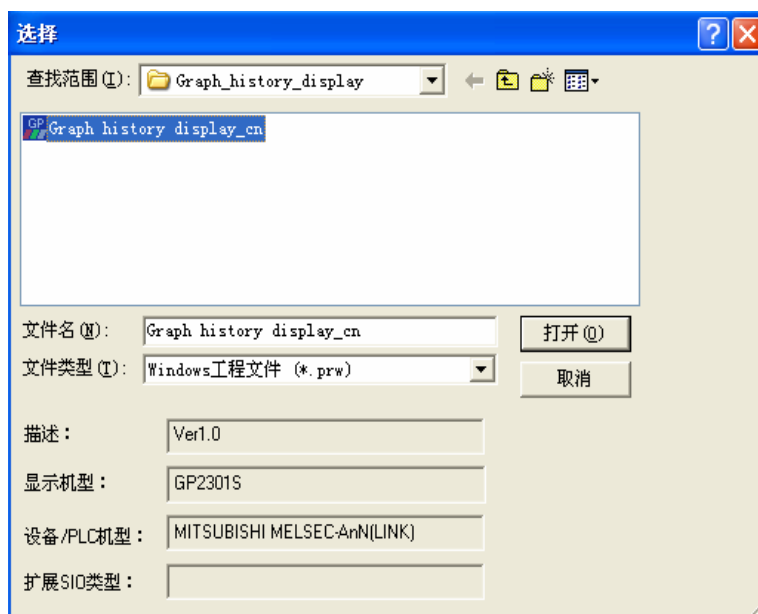
您可以将本例复制并安装到您已建立的工程中。请注意检查拷贝的相应的地址和画面编号没有重复。

(→确认使用地址的方法)

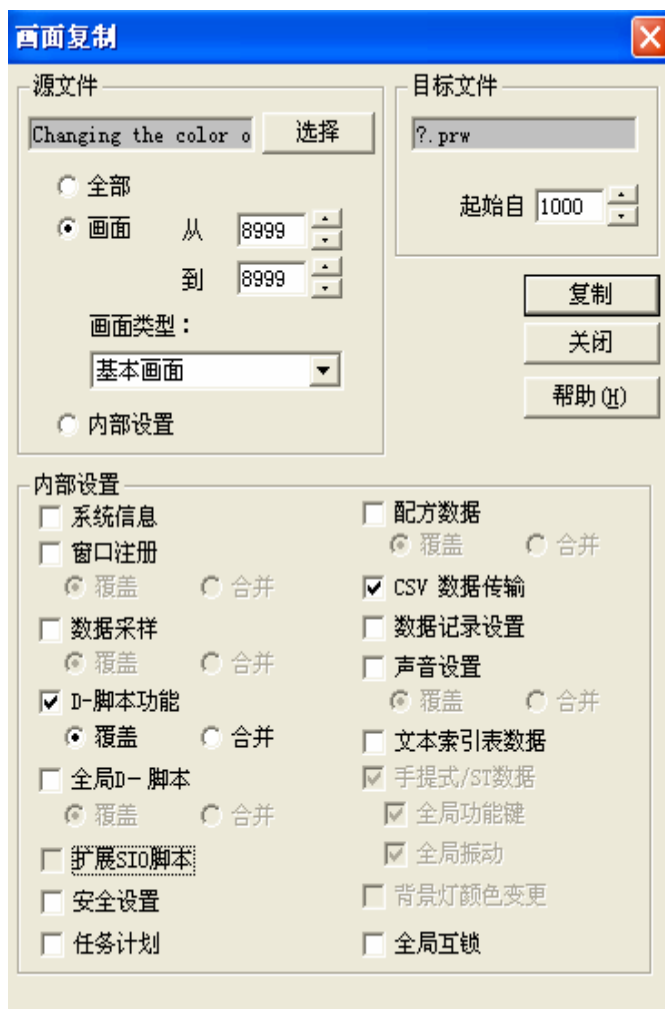
此外, 请根据您使用的触摸屏的型号, 适当调整画面的位置和相关设置。



在工程管理器的主菜单中, 点击[应用], 然后点击[画面复制]命令。



选择本例程序的文件名，然后单击[打开]。



在“画面”下，将“从”和“到”均设置为 8999，画面类型为“基本画面”，且设置想要复制到用户工程中的画面编号。（这里设置为“1000”）

在“内部设置”下，选择 D 脚本功能和“CSV 数据传输”。

设置完成之后，单击[复制]。

7. 注意事项

如果您使用的触摸屏型号与本例中的不一样（如画面尺寸变大），由于画面的位置不同，因此需要对画面进行修改。适用本例的是 GP2000 系列和 GLC2000 系列。使用的画面编辑软件版本是 6.1 以上。

本例需要 CF 卡，请使用 Pro-face 的 CF 卡（CA3-CFCALL/64MB-01, CA3-CFCALL/128MB-01, CA3-CFCALL/256MB-01, CA3-CFCALL/512MB-01）。

请确认在断电时，既没有图形显示也没有进行备份数据处理。（请参阅第 6 页的注释）

如果您使用的触摸屏通讯协议是“MEMORY LINK SIO Type”或是“MEMORY LINK Ethernet Type”，LS 区显示的方式也会不同。在安装本例时，部件的符号将自动转换，但 D 脚本中的符号保持不变，因此会发生错误。在这种情况下，请修改 D 脚本。

示例	PLC 连接方式	MEMORY LINK 连接方式
部件和 D 脚本的触发表示	LS1000	1000
D 脚本的操作表示	[b:LS100000] [w:LS1000]	[b:100000] [w:1000]

本例中，您可以使用 16 位的二进制数据（0~65535）。

如果使用了其它格式或负数数据，您需要在 D 脚本中进行相应的设置。

示例：使用 16 位二进制负数数据

D-脚本编辑器：历史显示

文件(F) 选项(O)

ID: 00002

描述: 历史显示

显示工具箱

公式:

```

if [ [w:LS2022] == 1 ]
{
  if [ [t:0087] > 300 ]
  {
    if [ [w:LS2409] == 0 ]
    {
      //在当前数据显示
      //在当前数据存储
      //当没有按历史显
    }
  }
}

```

触发: [E] [S] [R] [X] f>t t+f

动作: [w:LS2409] <> [t:0078]

函数: CF_Read, CF_Write, present_data

数据类型: Bin 数据长度: 16 位 符号 +/-

操作表达式正确

调用 新建 编辑 删除 复制 粘贴

确定 取消 帮助(H)

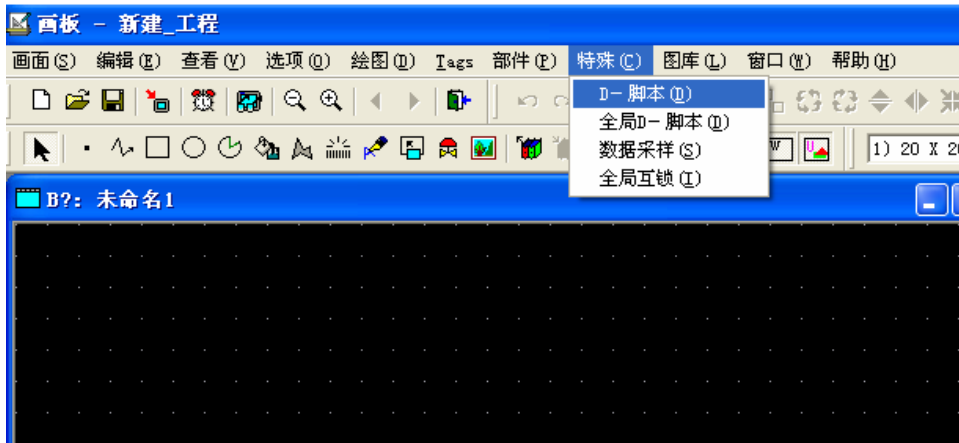
选择此项

在本例中，您可以使用以下数据（-32768~32767）。

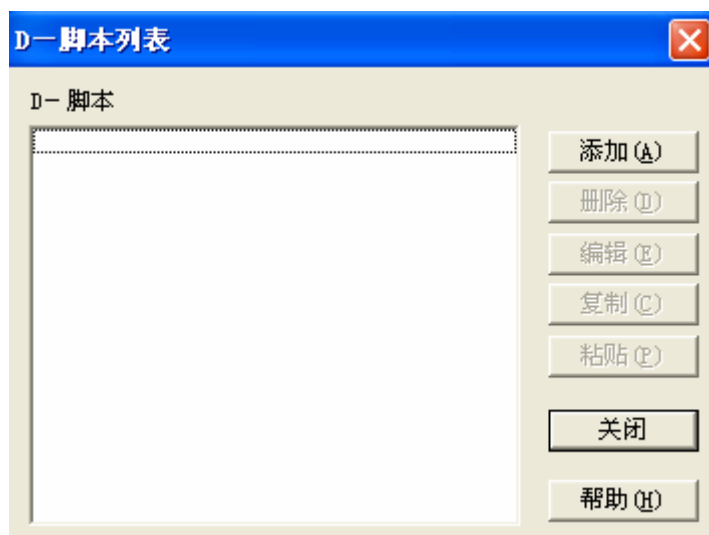
<附录>

建立和编辑 D 脚本

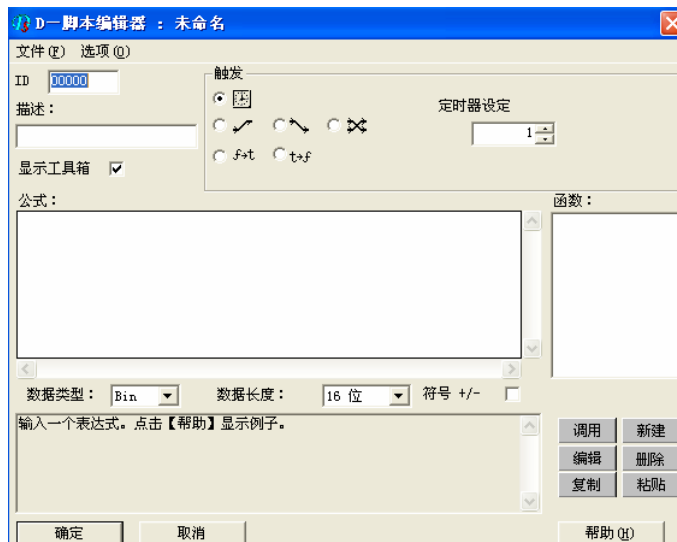
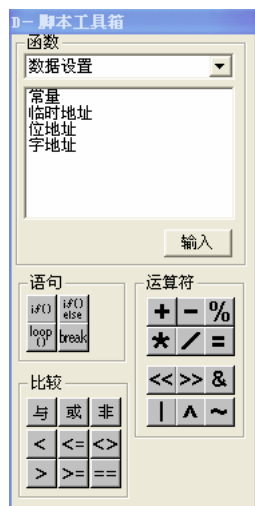
建立新的 D 脚本



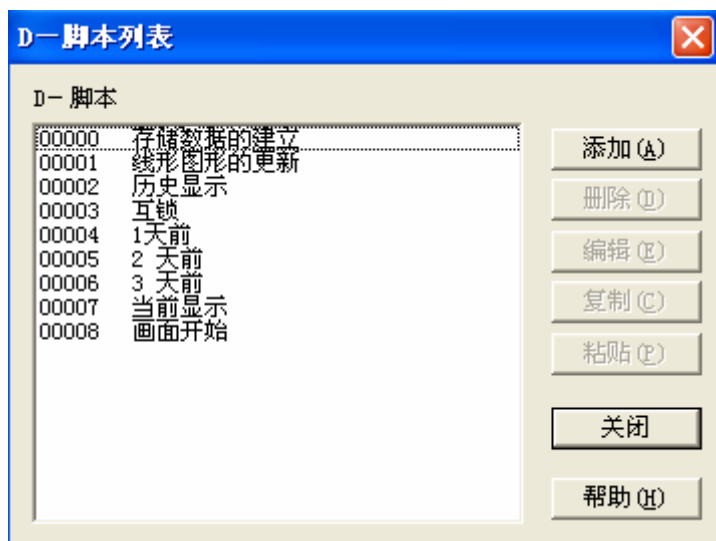
点击菜单栏上的[特殊]选项，在下拉菜单中选择[D 脚本]。



点击“D 脚本列表”中的[添加]。在显示 D 脚本编辑器后，使用“D 脚本工具箱”输入“地址”和里面的“运算符”。最后点击[确定]进行注册。



编辑 D 脚本

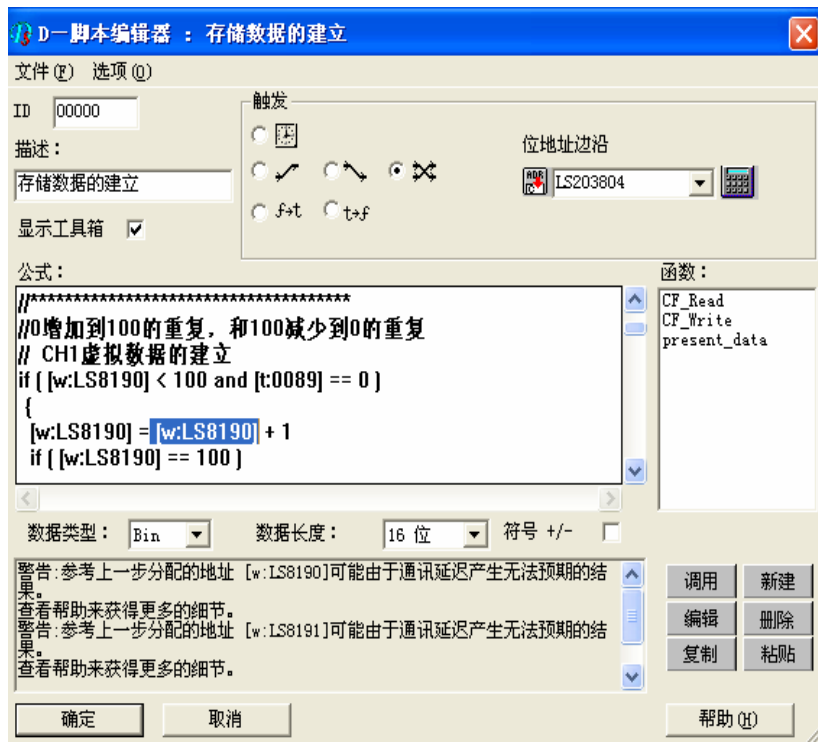


创建好的 D 脚本注册在脚本列表中。双击列表中的 D 脚本名称或点击[编辑]即可对其进行编辑。

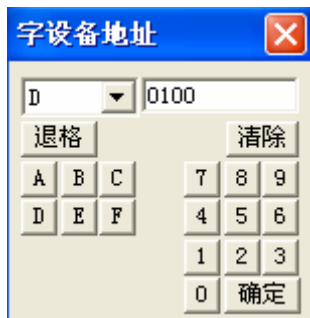


在显示所选择的 D 脚本后，编辑需要编辑的部分。然后点击“确定”进行覆盖。

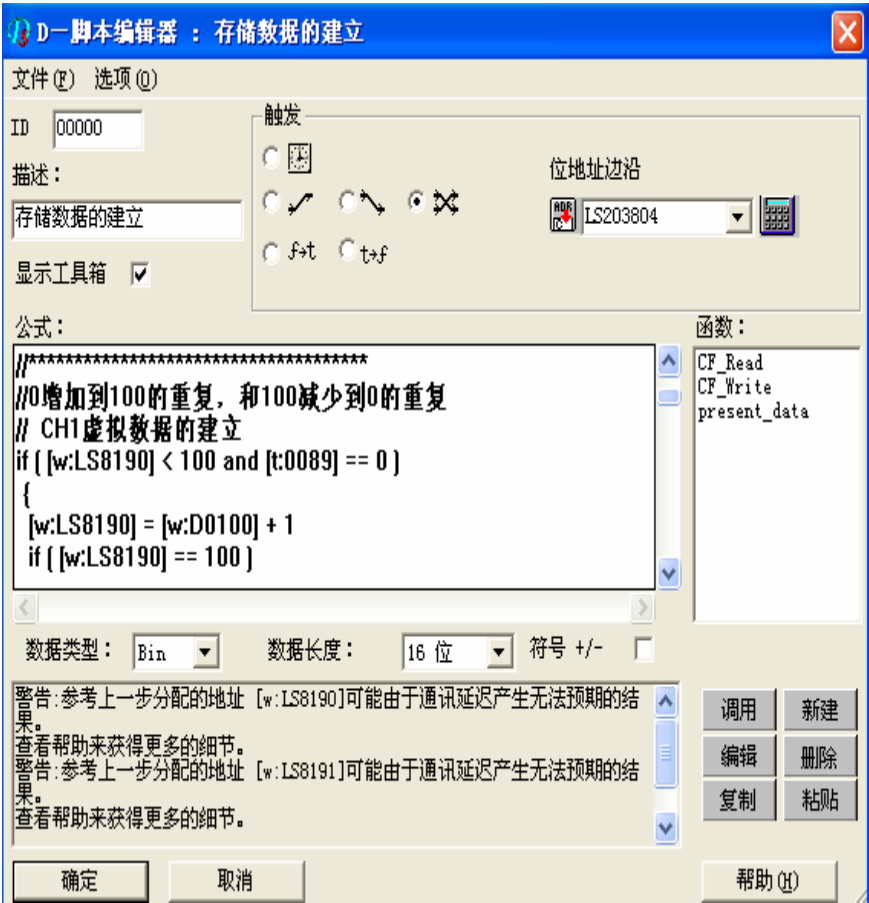
修改 D 脚本地址



打开您想修改的 D 脚本，双击您想修改的地址。
本例中，将“LS8190”修改为 PLC 中的“D100”。



输入“D0100”，
然后点击“确定”。



参照左边截图修改设备地址。

关于 LS 区

LS 区位于 GP 内存中，其地址分配如下：

LS 区的构成如下表：

LS0 : LS19	系统区
LS 20 : : : LS2031	用户区
LS2032 : LS2047	特殊继电器区
LS2048 : LS2095	保留区
LS2096 : : : : : : LS4095※	用户区

用户区是仅供 GP 内部使用的设备地址，不能在 PLC 内使用。这个区只能用于处理 GP 内部的部件和 Tag，PLC 不能控制。

（⇒参阅设备连接手册 1.1.2）

※ 对于 GP2000 系列，LS 区最大可以使用到 LS8191。

确认地址

下面部分解释了如何确认画面、D 脚本中所用的设备地址。

另外，也可以确认相应的画面号。



在工程管理器的[应用]菜单中，选择[全局交叉参考]-[列表]命令。

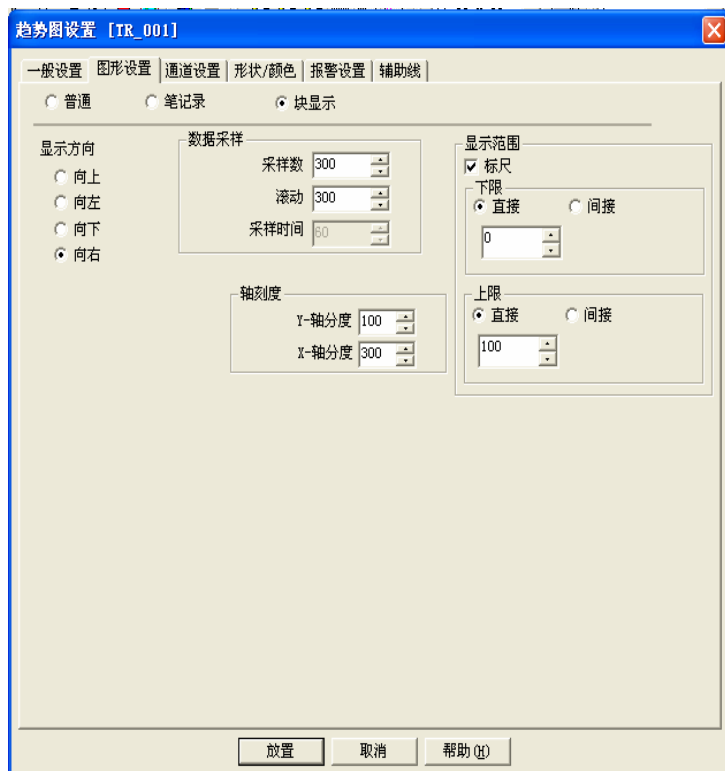


然后会弹出“全局交叉参考列表”窗口。列表中将显示已使用的地址和画面号。双击某个地址或点击窗口右侧的[打开画面]，输入您要查找的地址，然后使用该地址的画面将自动打开。

关于图形显示



点击趋势图图标（红圈内）。

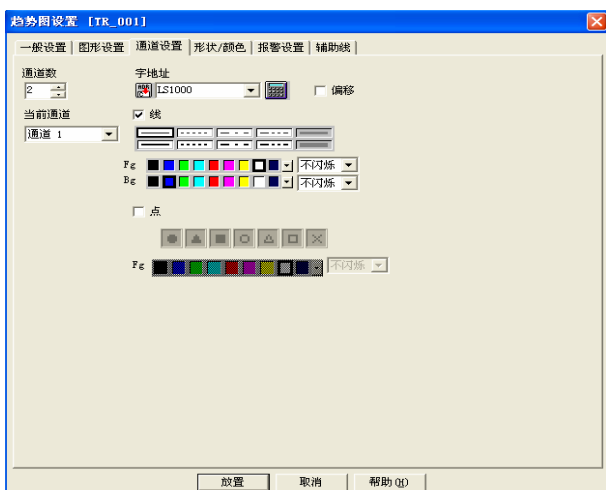
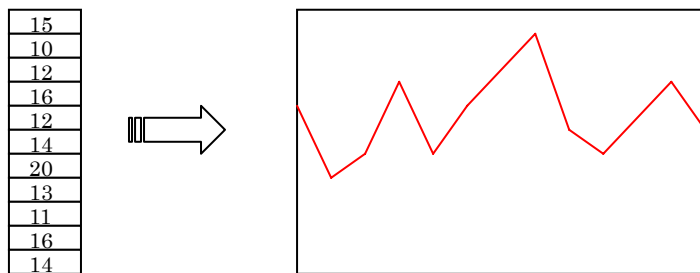


打开“趋势图设置”的“图形设置”标签。

图形类型：块显示*

数据采样：300（在图形中显示的标志）

※ “块显示”的功能是以图形的形式显示存储在 LS (PLC) 地址中的数据。



打开“通道设置”标签。

通道数：图形显示数。

字地址：设置所选通道的字地址。

※ 块显示的字地址

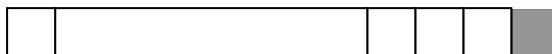
用于执行以下操作的块显示中的字地址区域。

LS1000	1	控制
LS1001	10	
LS1002	100	显示数据量
LS1003	300	
LS1004	600	
LS1005	400	
LS1006	200	
LS1007	300	
LS1008	500	
LS1009	200	
LS1010	100	
LS1011	400	

如果按左边的地址设置每一个数据并存储显示数据量后，将数据写入控制字地址，就会显示出图形。

关于控制

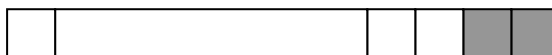
① 如果将 00 位设置为 1，则将执行图形显示。



② 如果将 01 位设置为 1，则将执行图形清除。



③ 在将 00 位和 01 位设置为 1 后，将清除并重新显示图形。

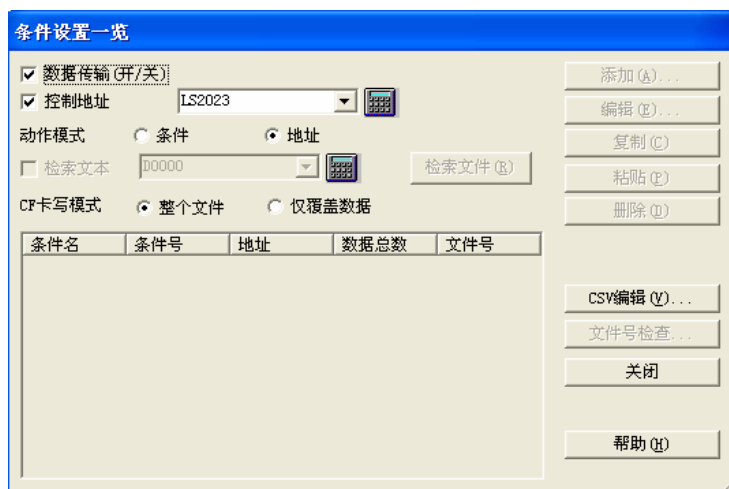


注意：在清除图形时，图形本身并没有消除，因为它已经被拷贝到了图形区域。如执行了 2 个或多个图形显示，则其它图形将消失。如要重新显示，需要一个程序。

关于 CSV 数据传输



点击[画面/设置]—[CSV 数据传输]—[条件设置]。



打开条件设置一览，选择“数据传输 (开/关)”，并设置一个字地址为控制地址。此外，“动作模式”项下选择“地址”。

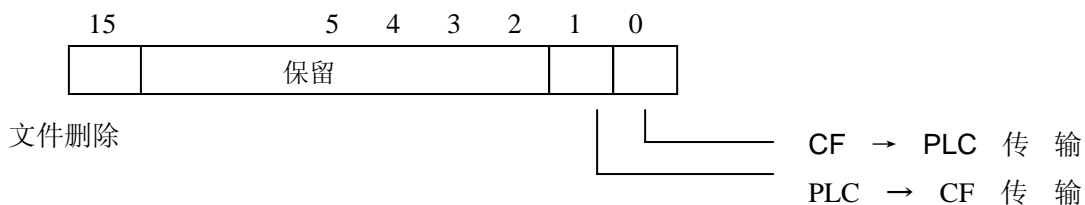
每一个 CSV 传输地址的操作设置都与下面的一样。

如果设置了要传输的文件编号、模式、设备代码、设备地址和数据量，且设置了控制值，那么将在 CF \leftrightarrow PLC (GP)之间执行数据传输。(→详情请参见 Tag 参考手册 4-4-6

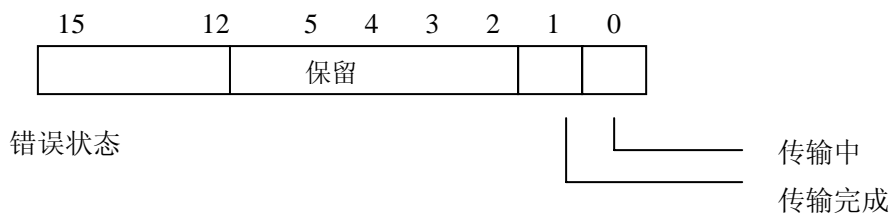
CSV 数据传输功能：关于地址操作中的传输控制地址)

LS2023	控制	CF \leftrightarrow PLC的触发位
LS2024	状态	传输完成状态
LS2025	文件编号	传输文件编号
LS2026	模式	传输数据格式
LS2027	设备代码	传输设备代码
LS2028	设备地址	传输设备地址
LS2029	数据量	传输数据量

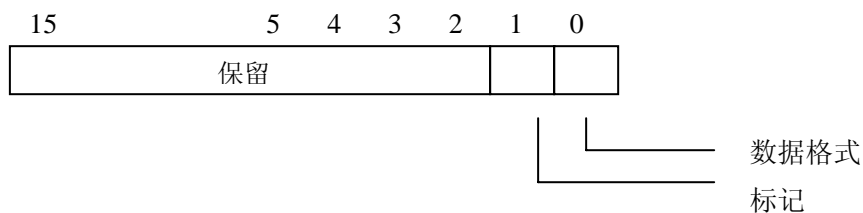
控制



状态



模式



关于 T-tag

T-tag: 此触摸屏开关用于写入 PLC 内部字地址。

- 模式

选择位/字/特殊之一。

- 位地址

触摸此位按钮时，该位地址将进行操作。

- 位访问

选择触摸按钮时进行的操作。

