

利用 **CF** 卡文件控制功能读写 **CSV** 文件 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. 画面操作方法.....	5
2.1 写寄存器画面（B: 8991）.....	5
2.2 CSV数据编辑画面（B: 8992）.....	8
3. 使用的地址.....	10
3.1 位地址.....	10
3.2 字地址.....	11
4. D脚本说明.....	12
4.1 “CSV数据读取和D寄存器存储画面” D脚本.....	12
4.2 “CSV 数据编辑画面” D脚本.....	13
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
8. <附录>.....	21
8.1 建立和编辑D脚本.....	21
8.2 关于LS区.....	24
8.3 确认地址.....	25

注意：在您的系统中使用本例时，在操作前请检查。

Excel 和 **ExcelVBA** 是美国微软公司的注册商标。

1. 概述

本例中使用了 CF 卡上的 CSV 文件作为 PLC 的配方。

因为 CSV 文件是文本文件，因此您可以使用电脑上的任何电子制表软件对其进行编辑，比如记事本和 Excel。此外，还可以通过操作触摸屏来改变 CSV 数据中的数据。这对于在电脑上编辑和管理 PLC 配方数据十分有用。

注意：在执行本例之前，需要复制 CF 卡上的随附文件夹。

<硬件配置>



※ CF 卡是必须的。

<操作概要>

有 2 个画面，其中一个用于将 CSV 配方数据写入 PLC，另一个用于编辑 CSV 文件的数据。

将数据写入 PLC:

- 指定工作号
- 下一步，点击寄存器存储按钮。
- 对应于工作号的一行 CSV 数据将被写入到 PLC 寄存器中。

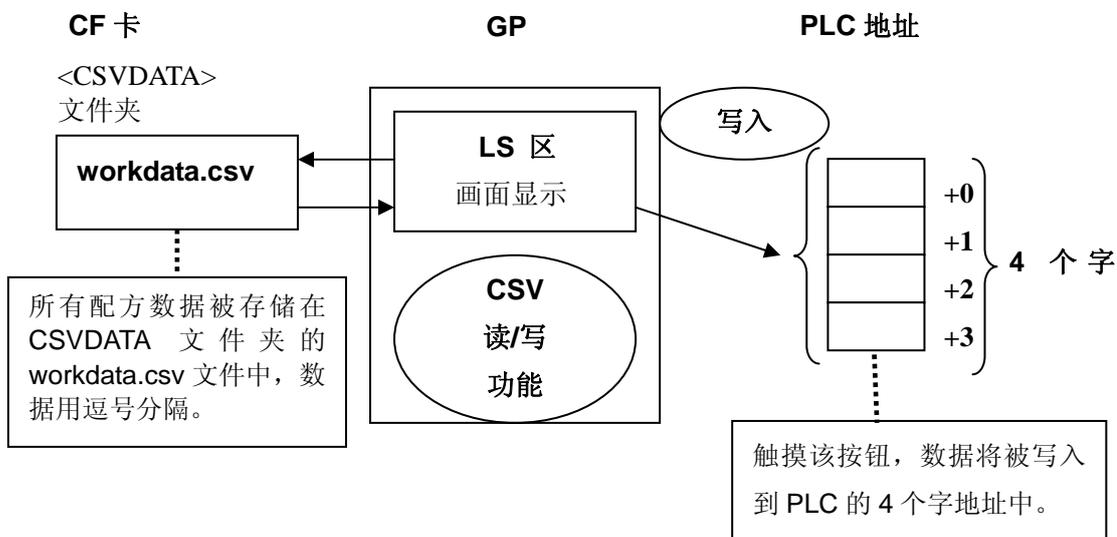
编辑 CSV 数据:

- 点击“CSV 数据读入”按钮，CF 卡中的 CSV 文件数据将被读出，并显示在画面中。
- 点击“CSV 数据写入”按钮，画面中的 CSV 文件数据将被保存到 CF 卡中。
- 点击“数据“0”-清除”按钮，画面上显示的数据就被清 0。

文件存储文件夹为<CSVDATA>。

文件名为 **workdata.csv**。

<操作流程图>



(注意) 本例仅适用于 GP (GLC) 2000 系列机型。

(在使用时参考注意要点)

CSV 文件图像

	1	2	3	4	5
1	WORK-No.	TMR01	TMR02	TMR03	
2		11	12	13	41
3		12	22	14	15
4		13	23	15	16
5		14	24	16	17
6		51	25	17	81
7					
8					
9					
10					
11					
12					
13					
14					

在后面的说明中，CSV 文件数据是指“工作数据”。

在后面的说明中，
 CSV 数据行 (不包括标题行) 是指“工作号”。

2. 画面操作方法

2.1 写寄存器画面 (B: 8991)

通过触摸来指定工作号，配方数据将根据相应的工作号写入 PLC 寄存器中。



工作号/键盘输入显示 (设置内容)



设置一个存储从 CSV 文件中读出的工作号的部件 (行号)。触摸“键盘输入显示”时, 会显示一个键盘 (数字键盘), 而当触摸“ENT”键时, 工作号将被保存到 LS7000 地址中。

· 数据 1~4 / 数值显示 (设置内容)



指定对应于数据 1~4 的 PLC 地址。
数据分配

- 数据 1: D1000
- 数据 2: D1001
- 数据 3: D1002
- 数据 4: D1003

· “CSV 读入和寄存器存储” 位开关 (设置内容)



触摸该按钮，LS700100 位将置 ON。
D 脚本的“00000(CSV 读取)”程序
由 LS700100 位的置 ON 触发运行。

在 D 脚本中，数据从 CSV 文件中相
应的工作号一行读出，再写入从 PLC
D1000 开始的 4 个字地址中。

· “CSV 数据编辑画面” 功能开关 (设置内容)



点击该按钮，画面将
切换到下一页 CSV
数据编辑画面。

- 错误信息/消息显示器（设置内容）

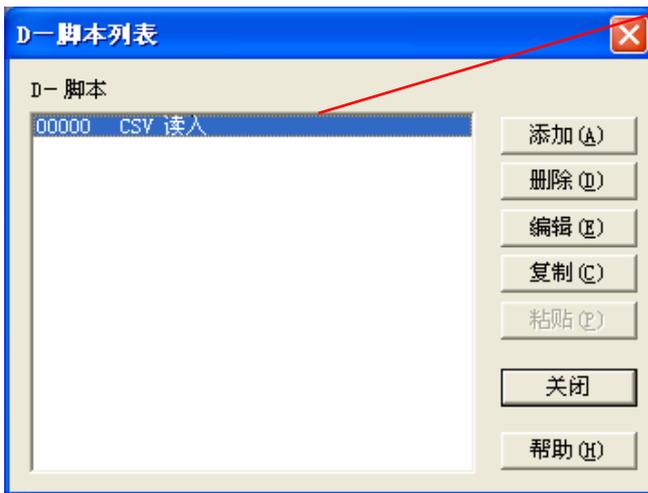


当在读取 CF 卡中 CSV 文件时发生错误，错误消息就会在消息显示上显示出来。

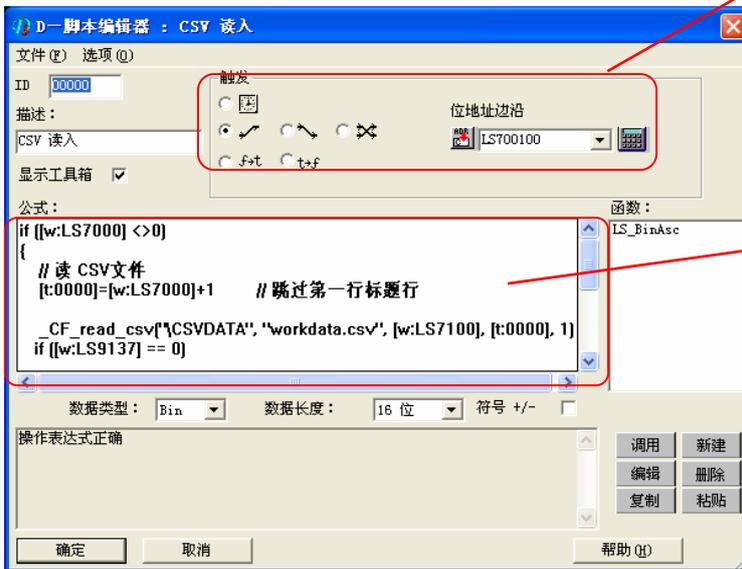
下面是显示的错误消息。

- 0: 空白
- 1: 参数错误
- 2: CF 卡错误
- 3: 写/读错误

- D 脚本



CSV 文件从 CF 卡读出，数据被写入 PLC。



设置执行 D 脚本的触发条件。
· 位上升沿触发
· “LS700100”

如果触发条件满足，这个 D 脚本将被执行。

2.2 CSV 数据编辑画面 (B: 8992)

(1) CF 卡中的 CSV 文件被读出并显示在画面上。

(2) 通过触摸修改 CSV 数据。

(3) 当修改完成后，触摸“CSV 数据写入”按钮，将其保存为 CSV 文件。

※ 即使没有读取任何数据，如果在创建 CSV 数据后触摸了“CSV 数据写入”按钮，将保存为一个新的 CSV 文件。

对右边显示表格中的数据清 0。

将 CSV 数据读入 GP 内部 LS

将 GP 内部 LS 区的数据写入 CSV 文件。

将画面切换到“CSV 数据读入和 D 寄存器存储画面”。

		工作数据			
		1	2	3	4
1号	数据“0”-清除	123	123	123	123
2号	CSV 数据读入	123	123	123	123
3号	CSV 数据写入	123	123	123	123
4号	D-寄存器写入画面	123	123	123	123
5号		123	123	123	123

在每个单元格中显示从 CSV 文件中读出的数值。
触摸后，会弹出一个十位键盘，用于修改数据。

- “数据“0”-清除”开关（设置内容）



触摸该按钮，LS700101 位将置 ON。当 LS700101 被置 ON 时，将执行 D 脚本“00002（数据清除）”。

在画面上显示的使用 GP 内部 LS 区地址的数据被清 0。

- “CSV 数据读入” 开关（设置内容）



触摸这该按钮，LS700102 将置 ON。

当 LS700102 被置 ON 时，将执行 D 脚本“00000（读 CSV 文件）”。

在 D 脚本中，CF 卡中 CSV 文件的数据被读入到 GP 内部 LS 区。

- “CSV 数据写入” 开关（设置内容）

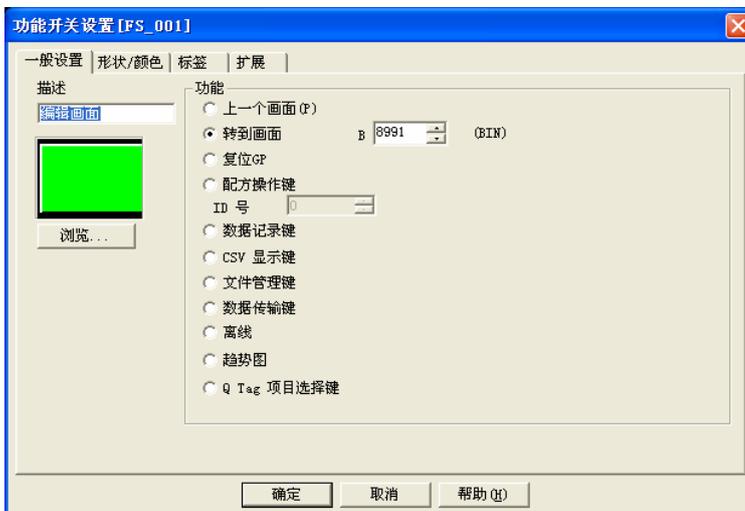


触摸该按钮，LS700103 将置 ON。

当 LS700103 被置 ON 时，将执行 D 脚本“00001（写 CSV 文件）”。

在 D 脚本中，GP 内部 LS 区的数据被写入 CF 卡并保存为 CSV 文件。

- “D 寄存器写入画面” 开关（设置内容）



触摸该按钮，画面将切换到前一幅“CSV 数据读入和 D 寄存器存储画面”。

3. 使用的地址

下面介绍本例中所使用的 LS 区和 D 寄存器的地址。(参阅关于 LS 区)

请根据您的系统更改这些地址。(参阅改变地址的方法)

3.1 位地址

LS7001		
15	---	
14	---	
13	---	
12	---	
11	---	
10	---	
9	---	
8	---	
7	---	
6	---	
5	---	
4	---	
3	点击“写 CSV数据写入”按钮	B2
2	点击“ CSV数据读入”按钮	B2
1	点击“数据 “0” -清除”按钮	B2
0	点击“CSV读入和存储寄存器”按钮	B1

3.2 字地址

D0000		系统设置
D1000	数据 1	B1
D1001	数据 2	B1
D1002	数据 3	B1
D1003	数据4	B1
LS5000	CSV文件写区域	B2
↓	↓	B2
LS5076	CSV 文件写区域	B2
LS6000	CSV 文件数据显示区域 1	B2
LS6001	CSV 文件数据显示区域 2	B2
LS6002	CSV 文件数据显示区域 3	B2
LS6003	CSV 文件数据显示区域 4	B2
LS6004	CSV 文件数据显示区域 5	B2
LS6005	CSV 文件数据显示区域 6	B2
LS6006	CSV 文件数据显示区域 7	B2
LS6007	CSV 文件数据显示区域 8	B2
LS6008	CSV 文件数据显示区域 9	B2
LS6009	CSV 文件数据显示区域 10	B2
LS6010	CSV 文件数据显示区域 11	B2
LS6011	CSV 文件数据显示区域 12	B2
LS6012	CSV 文件数据显示区域 13	B2
LS6013	CSV 文件数据显示区域 14	B2
LS6014	CSV 文件数据显示区域 15	B2
LS6015	CSV 文件数据显示区域 16	B2
LS6016	CSV 文件数据显示区域 17	B2
LS6017	CSV 文件数据显示区域 18	B2
LS6018	CSV 文件数据显示区域 19	B2
LS6019	CSV 文件数据显示区域 20	B2
LS7000	工作号	B1
LS7001	用作位地址	B1,B2
LS7100	CSV文件数据读存储区(数据 1)	B1
LS7101	CSV文件数据读存储区(数据 2)	B1
LS7102	CSV文件数据读存储区(数据 3)	B1
LS7103	CSV文件数据读存储区(数据 4)	B1
LS9134	CF卡写状态	B2
LS9137	CF卡读CSV状态	B1,B2

4. D 脚本说明

本例使用了画面编辑软件的 D 脚本功能。下面解释本例中 D 脚本的使用。解释 D 脚本之前先增加行号。（参阅[建立和编辑 D 脚本](#)）

D 脚本只能在当前画面中执行。而且只能在 GP 显示该画面时执行。

4.1 “CSV 数据读入和 D 寄存器存储画面” D 脚本

ID00000: CSV 读取

触发：当 LS700100 位被置 ON 时，将执行下面的内容。

```

01 if ([w:LS7000] <>0)
02 {
03     // 读取CSV文件
04     [t:0000]=[w:LS7000]+1    // 跳过第一行标题行
05     _CF_read_csv("¥CSVDATA", "workdata.csv", [w:LS7100], [t:0000], 1)
06     if ([w:LS9137] == 0)
07     {
08         // 复制到D寄存器
09         memcpy([w:D01000],[w:LS7100],4)
10     }
11     endif
12 }
13 endif

```

01: 判断 LS7000 地址中的数值是否大于 0 并判断工作号是否从 1 向上递增。将 CSV 文件第二行对应于工作号 1。CSV 文件第一行为标题行。

04: 1 被加为工作号。

（CSV 文件数据行从第二行开始。第一行为标题行。）

05: 从 CF 卡 CSV 文件中一行一行读数据。读出的数据被写入 LS7100 开始的地址中。

06: 判断 CSV 文件读取时有没有错误。比如，如果 GP 中没有插入 CF 卡，将产生一个错误。

09: 从 LS7100 开始的 4 个字地址中的数据被传送到 D1000 开始的寄存器中。

4.2 “CSV 数据编辑画面” D 脚本

ID00000: 读取 CSV 文件

触发: 当 LS700102 位被置 ON 时, 将执行下面的内容。

```
01 // 读取CSV文件
02 _CF_read_csv("¥CSVDATA", "workdata.csv", [w:LS6000], 2, 5)
```

02: 读出 CSV 文件中第二行到第五行的数据保存到 LS6000 开始的地址中。

ID00001: 写入 CSV 文件

触发: 当 LS700103 位被置 ON 时, 将执行下面的内容。

```
01 [t:0086]=0 // 字数
02 [t:0087]=0 // 项目数
03 [t:0089]=5 // 行数
04
05 // 标题行
06 [w:LS5000]=0x2257 // "W
07 [w:LS5001]=0x4F52 // OR
08 [w:LS5002]=0x4B2D // K-
09 [w:LS5003]=0x4E6F // No
10 [w:LS5004]=0x2E22 // ."
11 [w:LS5005]=0x2C22 // ,"
12 [w:LS5006]=0x544D // TM
13 [w:LS5007]=0x5230 // R0
14 [w:LS5008]=0x3122 // 1"
15 [w:LS5009]=0x2C22 // ,"
16 [w:LS5010]=0x544D // TM
17 [w:LS5011]=0x5230 // R0
18 [w:LS5012]=0x3222 // 2"
19 [w:LS5013]=0x2C22 // ,"
20 [w:LS5014]=0x544D // TM
21 [w:LS5015]=0x5230 // R0
22 [w:LS5016]=0x3322 // 3"
23 [t:0086]=17 // 字数
```

```
24
25 loop([t:0089])
26 {
27     [t:0088]=4 //项目数
28     loop([t:0088])
29     {
30         // 原始数据转换
31         [t:0000]=[w:LS6000]#[t:0087]
32         Call LS_BinAsc // Bin数据转换为ASCII码
33         // 转换结果替换
34         [w:LS5000]#[t:0086]=[t:0010]
35         [w:LS5001]#[t:0086]=[t:0011]
36         [w:LS5002]#[t:0086]=[t:0012]
37         // 终端码的寄存器
38         if ([t:0087] <> 0)
39         {
40             if (([t:0087] % 4) == 0)
41             {
42                 // 字符行
43                 [w:LS5000]#[t:0086]=0x0d0a
44             }
45             else
46             {
47                 // 划界字符
48                 [w:LS5000]#[t:0086]=0x2c30
49             }
50             endif
51         }
52         else
53         {
54             // 第一行(标题行)
55             [w:LS5000]#[t:0086]=0x0d0a
56         }
57         endif
58         [t:0087]=[t:0087]+1 // 项目数的增加
59         [t:0086]=[t:0086]+3 // 字节数
60     }endloop
61 }endloop
62 //在CF卡中保存文件
63 [t:0000]=154 // 120 = 6(字符数)*4(字符)=24(一行字符数)*5行
64 // +=34(标题数)
65 _CF_write("¥CSVDATA", "workdata.csv", [w:LS5000], 0,[t:0000], 0)
```

概要说明

- 01 ~ 03: 初始值设置
- 06 ~ 22: 标题行设置
- 25 ~ 61: 每个单元格的数值对应于一组字符串，逗号为分隔符。
- 63 ~ 65: 在 CF 卡中保存 CSV 文件。

详细说明

- 01: t: 0086 中置 0
- 02: t: 0087 中置 0
- 03: t: 0089 中置 5
- 06: ASCII 码 数值 0x2257("W)被设置到 LS5000。
- 07: ASCII 码 数值 0x4F52(OR)被设置到 LS5001。
- 08: ASCII 码 数值 0x4B2D(K-)被设置到 LS5002。
- 09: ASCII 码 数值 0x4E6F(No)被设置到 LS5003。
- 10: ASCII 码 数值 0x2E22(.)被设置到 LS5004。
- 11: ASCII 码 数值 0x2C22(,)被设置到 LS5005。
- 12: ASCII 码 数值 0x544D(TM)被设置到 LS5006。
- 13: ASCII 码 数值 0x5230(R0)被设置到 LS5007。
- 14: ASCII 码 数值 0x3122(1")被设置到 LS5008。
- 15: ASCII 码 数值 0x2C22(,)被设置到 LS5009。
- 16: ASCII 码 数值 0x544D(TM)被设置到 LS5010。
- 17: ASCII 码 数值 0x5230(R0)被设置到 LS5011。
- 18: ASCII 码 数值 0x3222(2")被设置到 LS5012。
- 19: ASCII 码 数值 0x2C22(,)被设置到 LS5013。
- 20: ASCII 码 数值 0x544D(TM)被设置到 LS501。
- 21: ASCII 码 数值 0x5230(R0)被设置到 LS5015。
- 22: ASCII 码 数值 0x3322(3")被设置到 LS5016。
- 23: t: 0086 设为 17 (数值被转换为字符串)。
- 24: t: 0089 中设置的数值是循环的。第 5 行 CSV 文件循环。
- 27: t: 0088 设为 4。
- 28: t: 0089 中设置的数值是循环的。第 4 行 CSV 文件循环。
- 31: 在 t: 0000 中设置原二进制转 ASCII 码的数值。
- 32: 调用二进制转 ASCII 码的功能。
- 34: 设置二进制转 ASCII 的结果(1,00,000 和 10,000 位数)。
- 35: 设置二进制转 ASCII 的结果(1000 和 100 位数)。
- 36: 设置二进制转 ASCII 的结果(10 and 1 位数)。

- 38: 判断 t: 0087 是否为 0。(判断是否是第一行)
- 39 ~ 51: 当 t: 0087 数值超过 0, 程序完成。(程序运行除了第一行)
- 53 ~ 56: 当 t: 0087 数值为 0, 程序完成。(运行第一行的程序)
- 40: 是否是这一行的最后一个项目(第 4 个项目)由 t: 0087 决定。
- 43: 在一行的最后一个项目(第 4 个项目)后, 设置提示字符(CR + LF)。
- 48: 在一行的每个项目之间(第 1~3 个项目), 设置提示字符(CR + LF)。
- 55: 在第一行设置提示符(CR + LF)。
- 58: 1 被加入 t: 0087(项目数)。
- 59: 3 被加入 t: 0086(存储目的单元格)。
- 63: 设置 CSV 文件的字节总数。
- 65: 使用 CF_write()功能, 将文件保存在 CF 卡中。

ID00002: 数据清除

触发: 当 LS700101 被置 ON 时, 将执行下面的内容。

```
01 memset([w:LS6000],0,20) // 清除显示的数据
02 [w:LS9137]=0 // 清除CF_read_csv()功能错误
03 [w:LS9134]=0 // 清除CF_write()功能错误
```

- 01: 清除显示的 LS 区数据。
- 02: 清除 CF_read_csv()功能错误。
- 03: 清除 CF_write()功能错误。

LS_BinAsc

```

01 //
02 // LS的二进制数据转换成 ASCII码
03 //   输入:
04 //       [t:0000]:转换前数值
05 //   输出:
06 //       [t:0010]:转换成 ASCII后的数值(10000,      )
07 //       [t:0011]:转换成 ASCII后的数值(   100, 1000)
08 //       [t:0012]:转换成 ASCII后的数值(     1,   10)
09 //
10 //
11
12 // 分析每位数
13 [t:0005]=([t:0000]      ) / 10000
14 [t:0004]=([t:0000] % 10000) / 1000
15 [t:0003]=([t:0000] % 1000 ) / 100
16 [t:0002]=([t:0000] % 100  ) / 10
17 [t:0001]=([t:0000] % 10   ) / 1
18 // 显示的数据
19 [t:0010]=((          0x30) << 8) | ([t:0005] | 0x30)
20 [t:0011]=(([t:0004] | 0x30) << 8) | ([t:0003] | 0x30)
21 [t:0012]=(([t:0002] | 0x30) << 8) | ([t:0001] | 0x30)

```

- 13: 在 t: 0005 中放置第 10,000 的数值。
- 14: 在 t: 0004 中放置第 1000 的数值。
- 15: 在 t: 0003 中放置第 100 的数值。
- 16: 在 t: 0002 中放置第 10 的数值。
- 17: 在 t: 0001 中放置单元的位数。
- 19: 以 16 位数据存储的第 10,000 的数值被转换为 ASCII 码。
- 20: 以 16 位数据存储的第 100 和第 1000 的数值被转换为 ASCII 码。
- 21: 以 16 位数据存储的第 10 的数值被转换为 ASCII 码。

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。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。（参阅 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）

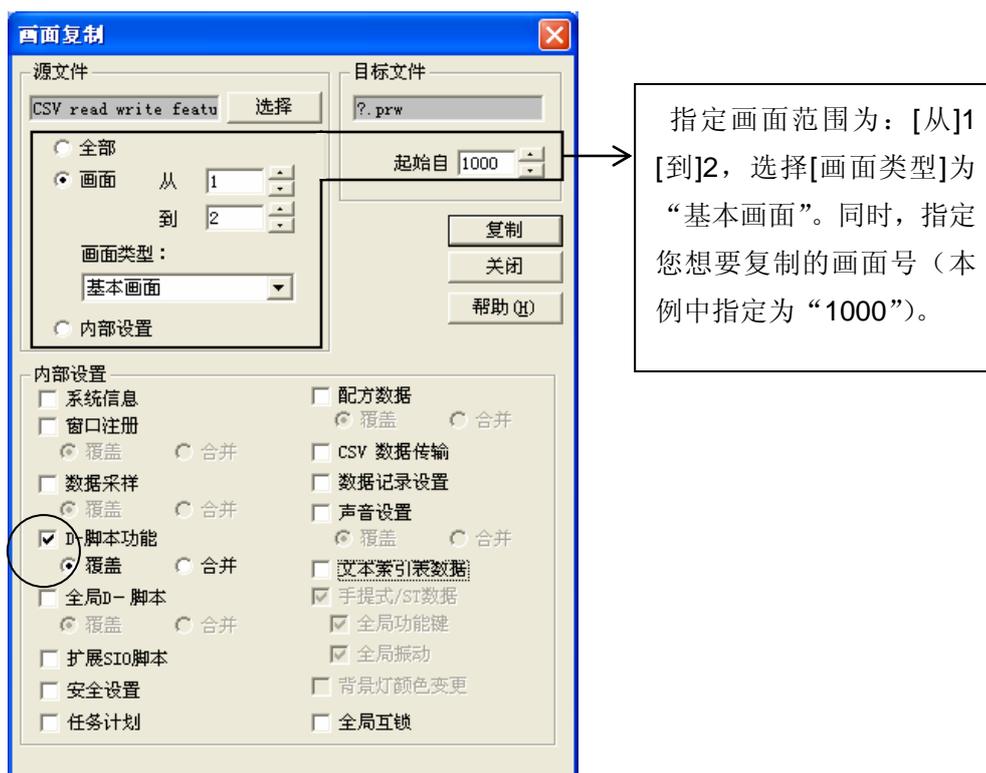
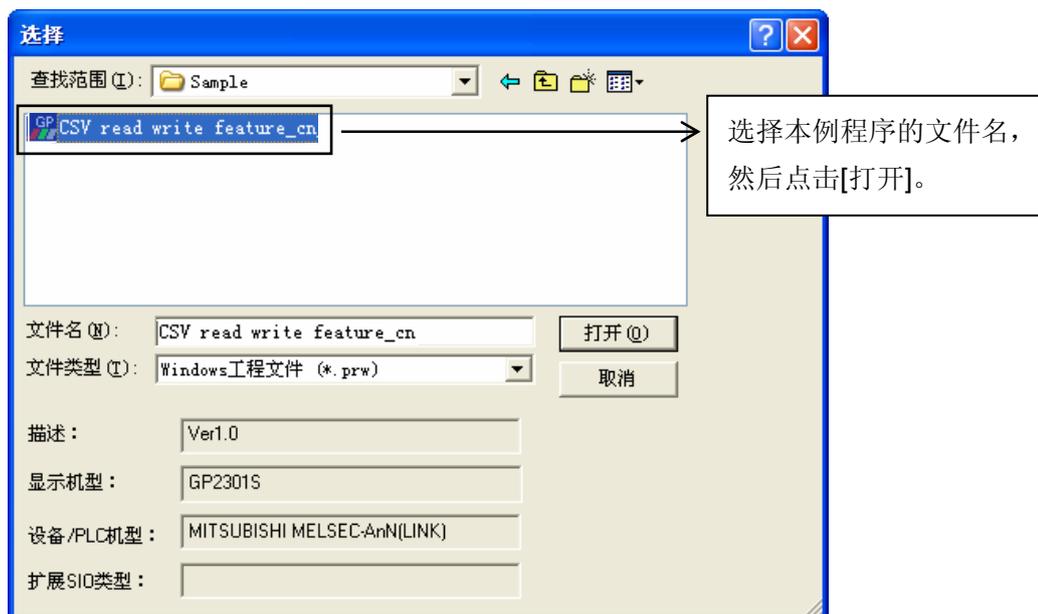
6. 画面复制

您可以将本例复制并安装到您已建立的工程文件（画面文件）中。需要注意的是已经使用的地址和画面号不能重复。（参阅确认地址）

同样，您需要根据所选的触摸屏类型适当调整画面的位置和设置。



在工程管理器的主菜单中，点击[应用]，然后再点击[画面复制]命令。



※ 如果您不希望使用 D 脚本的[覆盖]功能，请选择[合并]。

7. 注意事项

如果您使用的触摸屏型号与本例中的不一样（如您的触摸屏尺寸不同于本例中的画面尺寸），那么画面设置也会不同。因此，您可能需要调整画面设置。所选择的触摸屏的功能需与本例中的相同，均为 GP2000 系列和 GLC2000 系列。所使用的画面编辑软件版本为 7.0 以上。

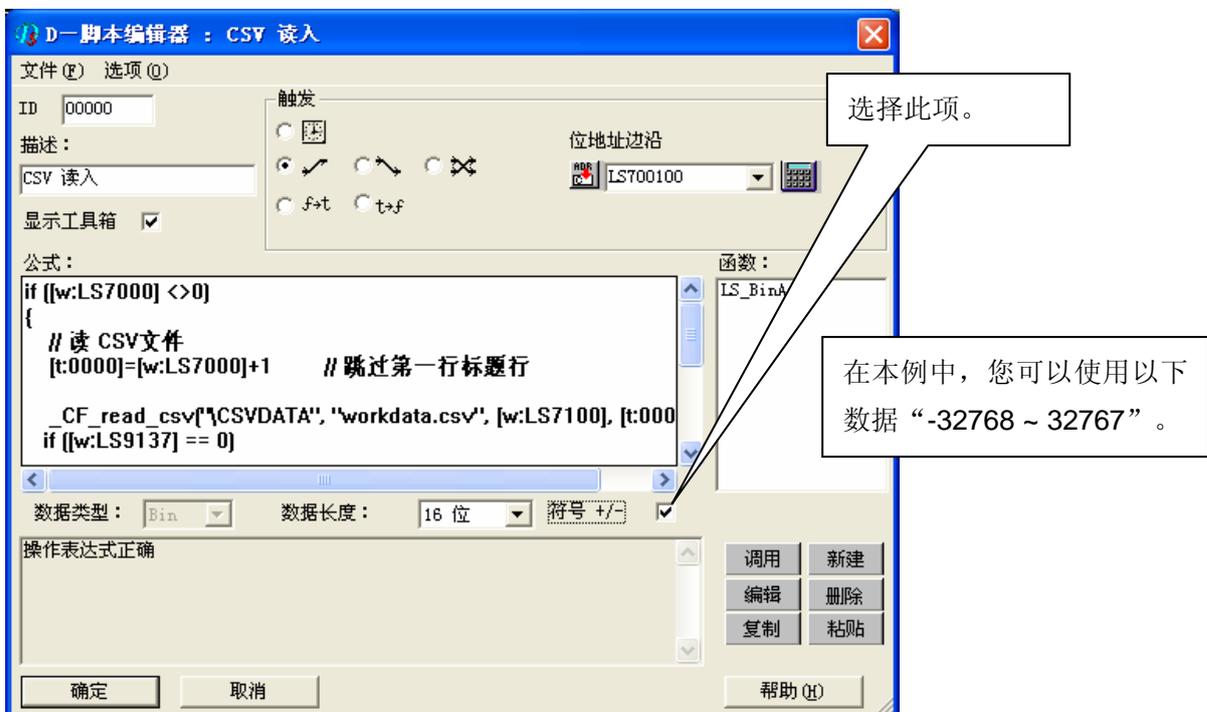
如果您使用的触摸屏通讯协议是“MEMORY LINK SIO Type”或是“MEMORY LINK Ethernet Type”，LS 区显示的方式也会不同。在安装本例时，部件的符号将自动转换，但 D 脚本中的符号保持不变，因此会发生错误。在这种情况下，请修改 D 脚本。

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 到 65535）。

但是，如果使用了负数数据或其它格式，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



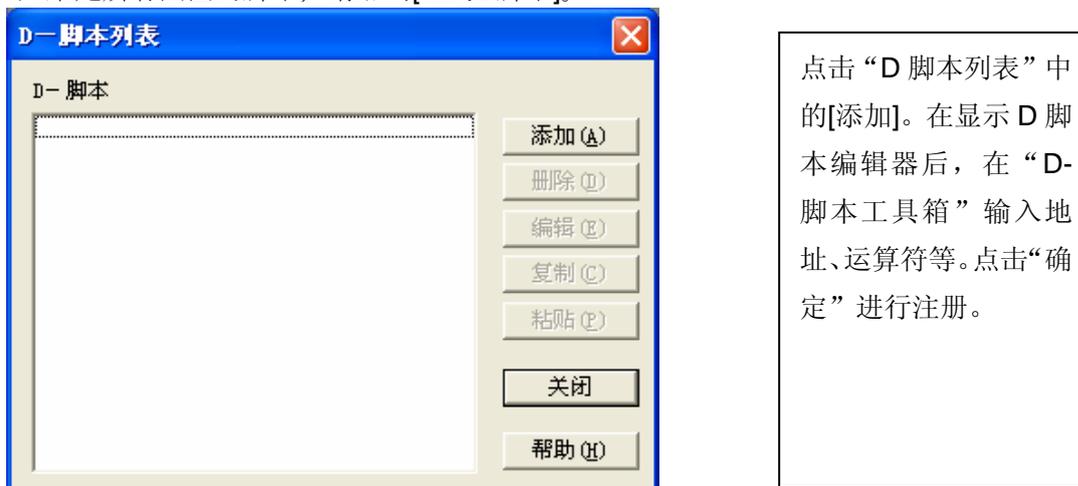
8. <附录>

8.1 建立和编辑 D 脚本

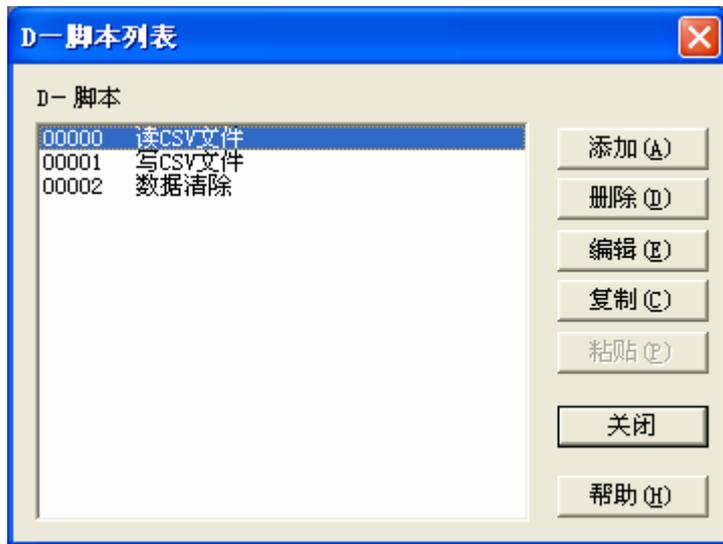
建立新的 D 脚本。



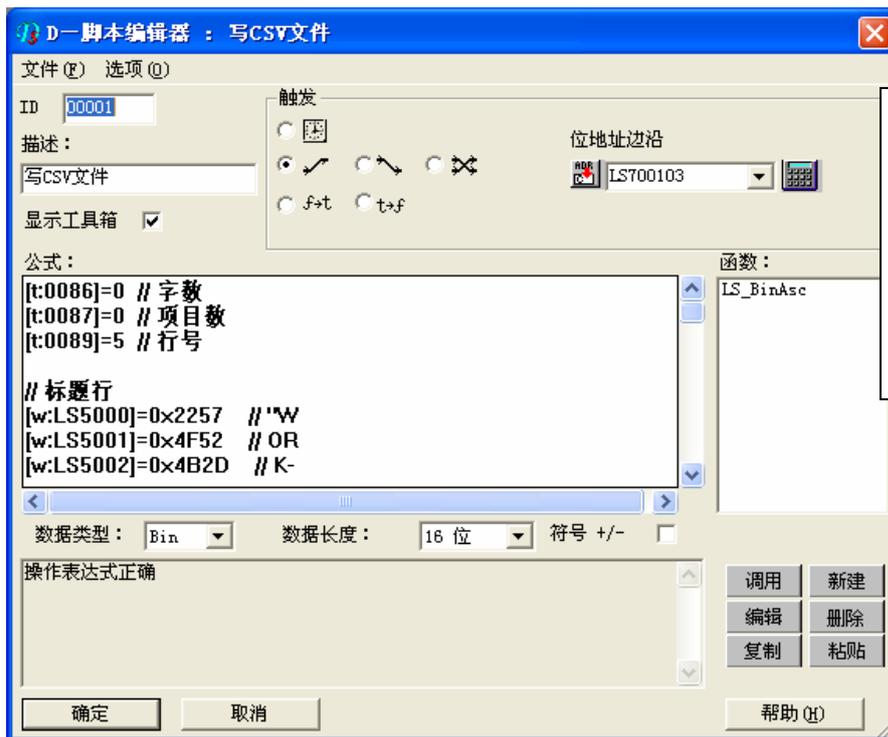
*如果是所有画面的脚本，请点击[全局D脚本]。



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。

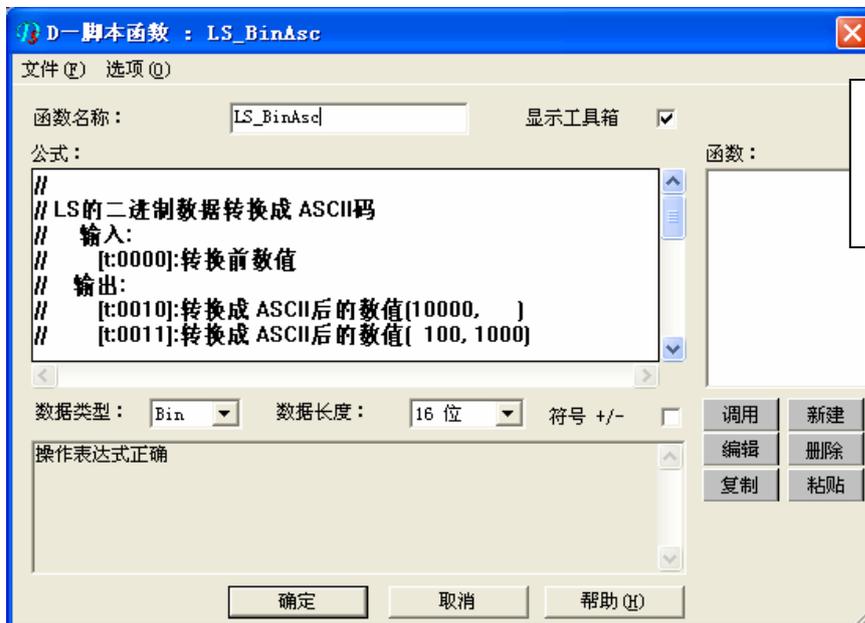


显示所选择的 D 脚本。然后编辑需要修改的部分。最后点击“确定”保存数据。

使用 D 脚本的函数功能



创建好的 D 脚本函数注册在 D 脚本函数列表中。如果需要编辑某个 D 脚本函数，请选择该该函数并点击“编辑”或直接双击该函数名称。



显示所选择的 D 脚本函数，然后编辑需要修改的部分。最后点击“确定”保存数据。

8.2 关于 LS 区

LS 区位于 GP 内存中，其结构如下所示。



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。

(→参阅“设备/PLC 连接手册”， 1-1-2)

※对于 GP2000 (GLC2000) 系列，LS 区最大可以使用到 LS8191。

8.3 确认地址

以下部分解释了如何确认当前用于画面或 D 脚本的设备地址。此外，也可以用同样的方法确认设备地址所使用的画面号。



在工程文件中，选择工程
管理器的[应用]菜单，然后
点击[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[打开画
面]并输入地址，这样就可以
打开该画面了。