

## 32 “位”字符和数据记录 说明书

普洛菲斯国际贸易（上海）有限公司  
技术热线：021-6361-5008

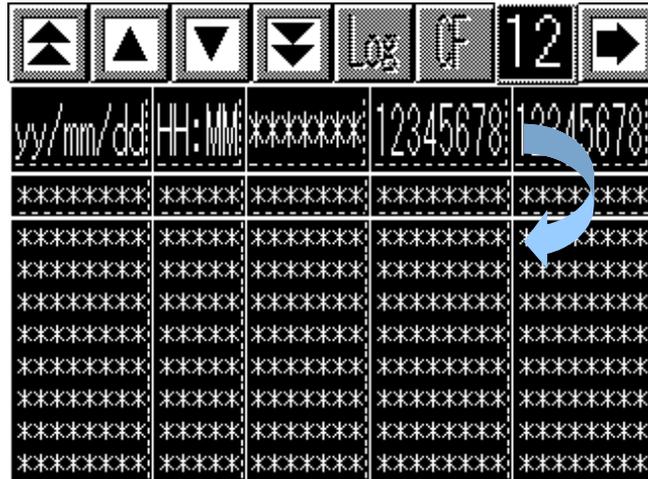
## 目录

内容	页码
1. 概述.....	3
2. 画面建立 .....	4
3. 地址和 D 脚本说明.....	13
4. GP、PLC 和画面编辑软件版本.....	20
5. 画面复制.....	20
6. 注意事项.....	22
 <附录>	
建立和编辑 D 脚本.....	23
修改 D 脚本地址.....	25
关于 LS 区.....	26
确认地址.....	27
编辑弹出键盘.....	28

注意：在您的系统中使用本例时，在操作前请检查。

## 1. 概述

下列方法要说明的功能是使用 D 脚本将字符串和 32 “位” 数据，通过 GP 以 CSV 文件格式存储到 CF 卡里。使用普通的数据记录功能是无法完成这类操作的。首先设置文件名和项目名，然后启动数据记录。如果数据记录达到 50 次，那些数据将以 CSV 文件格式自动存储到 CF 卡里。您也可以将 CF 卡内的 CSV 文件数据显示在画面上，以便于您对数据进行检查操作。



数据首先记录在 LS 区内



存储在 CF 卡里

Date	Time	Type	Units	Shipment
2004/6/9	19:37	PRO/PB3	38787558	534
2004/6/9	19:36	PRO/PB3	54454446	23235322
2004/6/9	19:36	GLC2500	66553554	54211222
2004/6/9	19:36	GLC2500	55557225	4776554
2004/6/9	19:35	GLC2500	12545886	4448887
2004/6/9	19:35	FGW	96556	65465
2004/6/9	19:34	GP-WEB	23362	2355
2004/6/9	19:34	GP-WEB	411556	225212

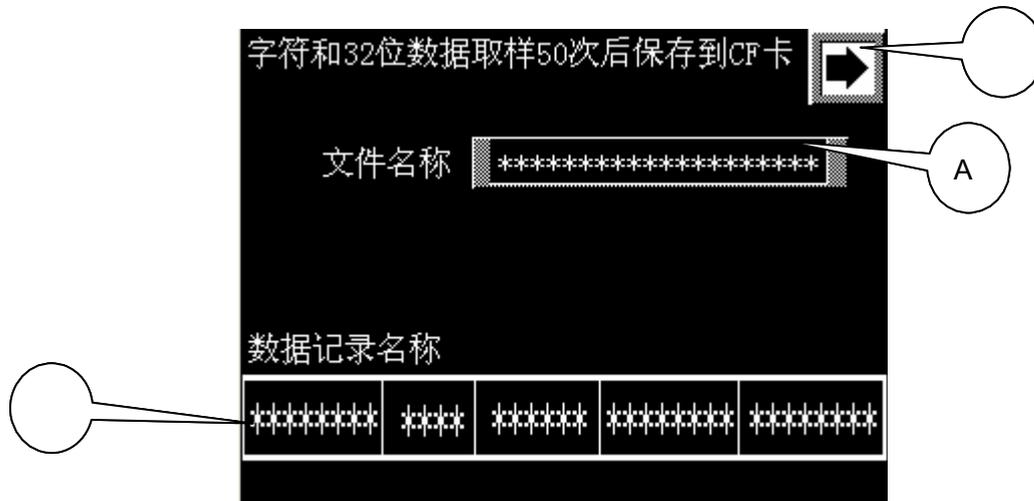
一个存储的 CSV 文件

注意：这个示例只能工作在 GP (GLC) 2000 系列产品上。

在实际使用中，这个写数据 CSV 文件的操作可能需要少许的时间来完成。（→6. 注意事项）

## 2. 画面建立

输入一个数据记录的项目名和一个存储器字符串文件的文件名。



### A. 数字显示设置

GP 上电正常运行后，输入一存储时使用的文件名（ZL00000.CSV）。可以通过触摸 A 来进行文件名的输入。同样也可以用 PLC 的地址完成文件名的设置，此类应用必须对部分 D 脚本进行修改。（→更改 D 脚本中使用的地址）



项目名和文件名是可以自由设置的，当使用 GP 或 Excel 等显示这些数据时，可以设置成“文件名.CSV”。

## B. 键盘输入显示

用于数据记录的项目名称输入和显示字符串类型的数据。在 GP 上电运行时，可以通过触摸它来改变编辑“日期”“时间”“字符串”“数据 1”“数据 2”并存储。在这里使用的地址可以改变成 PLC 的地址。您需要对 D 脚本做部分修改。（→改变 D 脚本中使用的地址）



\*设置键盘数据显示部件（A 和 B 的设置操作基本相同）



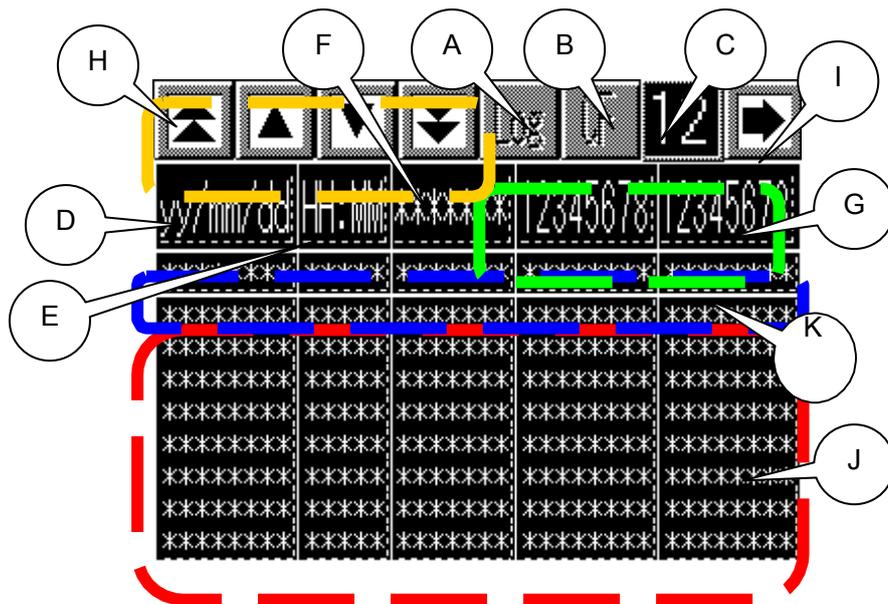
C. 功能开关（画面转换）使画面显示跳转到 B8992。



D. 在使用的 GP 型号为 GP2301 必须改变弹出的字符串输入键盘。（→编辑弹出键盘）

#### B8992 数据记录监控

输入数据记录的数据(字符串和 32 “位” 的 Dec 数据)并按下“Log”按钮, 于是数据被记录同时之前的数据下移一行。最新的数据总是显示在顶部。每当数据记录完成一次操作时, 数据记录计数器就会增加 1。当次数达到 50 次时, 数据自动的以设置的文件名保存 (在\Log 文件夹下)在 CF 卡里。此外, 每当“GF”按钮被按下时, 数据记录计数器的数值均会被保存。



### A. “位” 开关

这个按钮开关用于数据记录。功能是“置位”，使 D 脚本的触发地址变为 ON。也可以由 PLC 地址来完成触发，这种情况下必须去改变一部分 D 脚本。\*重新编辑 D 脚本。(→更改 D 脚本中使用的地址)

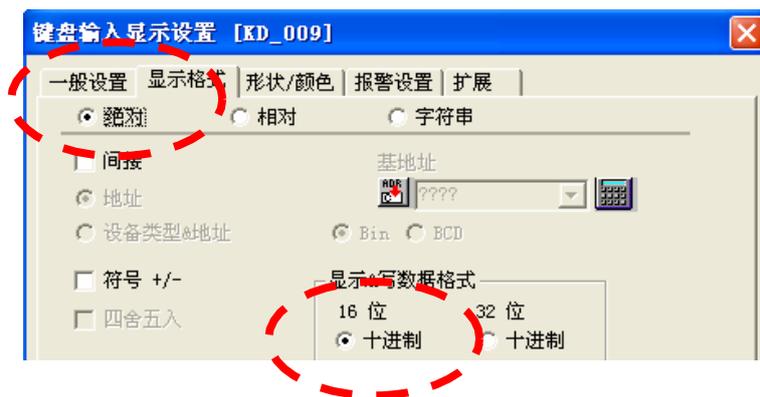
### B. “位” 开关

这个开关用于将数据保存到 CF 卡里。功能是置“位”，使 D 脚本的触发地址变为 ON。也可以由 PLC 地址来完成触发，这种情况下必须去改变一部分 D 脚本。\*重新编辑 D 脚本。(→更改 D 脚本中使用的地址)

### C. 设置数值显示

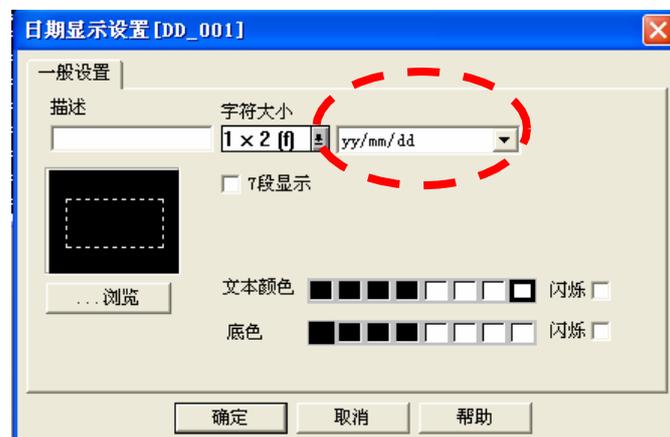
\* 数据记录的次数计数。当次数达到 50 次时，数据写入 CF 卡内，记录次数计数器数值自动清除为 0。(自动写操作)

\*按下那个“CF”按钮,当前显示的最新数据行的数据将保存到 CF 卡里。在保存前您也可以对这些数据进行编辑。(自动写操作)



### D. 日期显示

显示 GP 的日期数

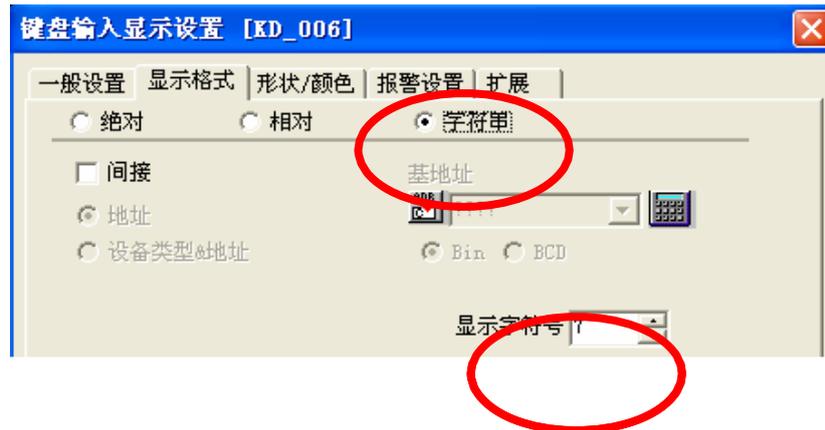


## E. 时间显示

显示 GP 的时间数

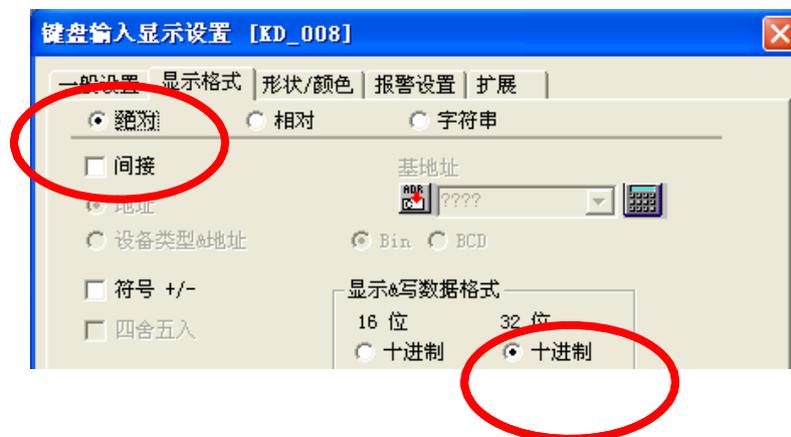
## F. 设置数值显示

用于数据记录的数据输入。也可以由 PLC 地址来完成触发，这种情况下必须去改变一部分 D 脚本。  
\*重新编辑 D 脚本。(→更改 D 脚本中使用的地址)



## G. 设置数值显示

输入希望被记录的 32 “位” Dec 数据。也可以由 PLC 地址来完成触发，这种情况下必须去改变一部分 D 脚本。  
\*重新编辑 D 脚本。(→更改 D 脚本中使用的地址)

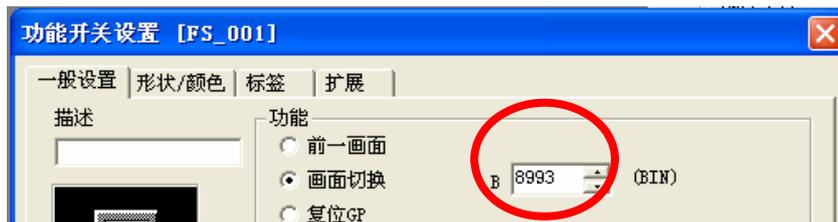


#### H. “位” 开关

这个开关可以对数据记录显示行进行上下移动操作。D 脚本触发“位”地址为 ON。功能为“瞬动”。

#### I. 功能开关(画面转换)

这个开关将显示跳转到 B8993 号画面。



#### J. 键盘输入显示设置

数据记录内容的显示。输入开始“位”是预先通过使用特殊区域定期关闭的“位”来完成的。当卷按钮按下时，添加数据到字地址，显示更多地址内的数据。(-关于LS区)



\*键盘输入显示设置，间接基础地址的配置。

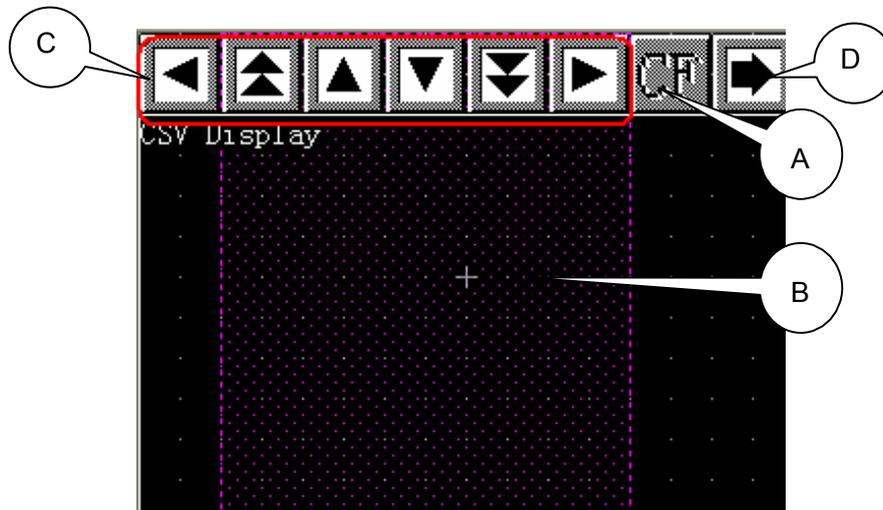
LS7022	LS7027	LS7030	LS7034	LS7039
LS7044	LS7049	LS7052	LS7056	LS7061
LS7066	LS7071	LS7074	LS7078	LS7083
LS7088	LS7093	LS7096	LS7100	LS7105

·            ·            ·            ·            ·  
·            ·            ·            ·            ·  
·            ·            ·            ·            ·

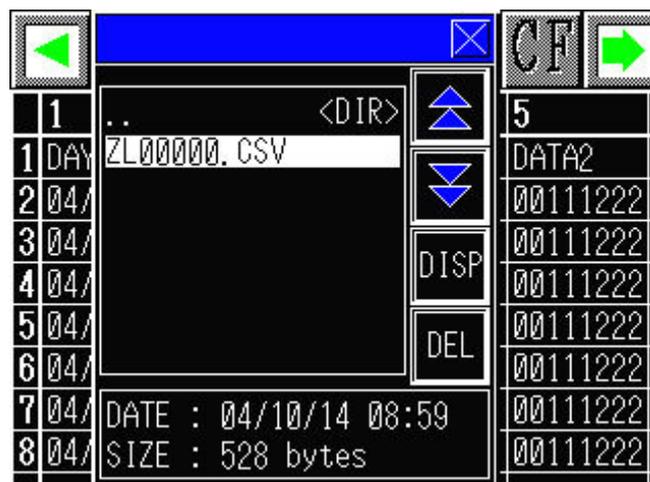
**B8993: CSV 显示**

保存在 CF 卡里的数据可以在 GP 的画面上显示出来。那个保存的数据存储在“/Log”文件夹里。

如果通过文件管理器选择了一个需要显示的文件，它将会在 CSV 文件显示器上显示出来。

**A. 文件管理器显示**

这是一个用于显示 CF 卡里的文件夹和文件目录的部件。当“CF”按钮按下后,将显示文件管理器窗口。如果选择“Log”按下“Display”然后选择 CSV 文件里的 ZL00000.CSV 文件(如下图),它的  
数据将显示在 CSV 文件显示器里。如果选择 CSV 文件后,按下“Delete”按钮,所选择的文件将被删除。



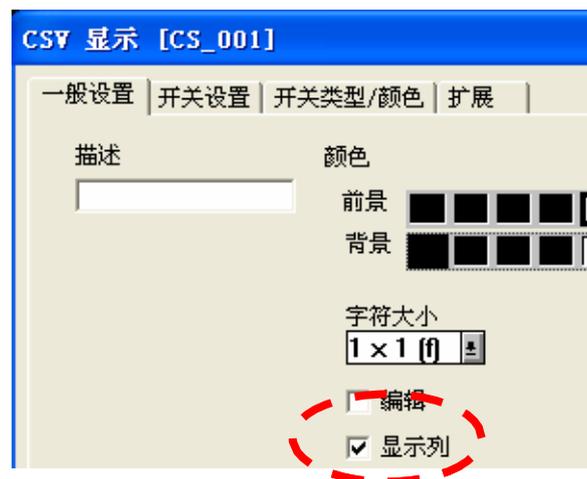
\*在 CSV 显示器上打开的文件是不能进行删除操作的。

关于“Display”和“Delete”按钮，通过在文件管理器，显示配置窗口的一般标签里，选择“包含删除按钮”和在开关设置标签里选择放置显示开关选项，来完成设置。此外，放置的这些部件最初是以组合部件形式显示在画面上的，如希望调整各个部件的位置，可以通过取消组合来进行各个部件的调整。



## B. CSV 显示

这个部件用于显示 CF 卡里 CSV 文件的内容。在放置时它是将各个部件组合后以组合部件方式放置的。如果取消组合,开关和显示区域的大小位置均可自由调整。如果按下“显示”按钮,选择的文件在文件管理显示器里显示出来,隐藏的部分文件通过卷的移动键来显示。



\*如果“显示列”激活,将在哪个列和行上将显示列和行的数量。如果“编辑”同样被激活,存储的 CSV 文件将可以进行编辑。

### C. 功能开关（显示键）

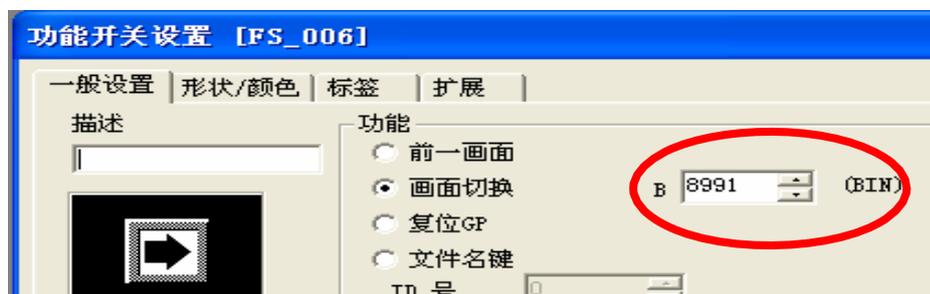
这个开关可以完成 CSV 文件卷的左右移动和上下移动，同时可以对移动的行列数量进行设置。



\*上面的图片是一个设置的例子，其完成卷右移一行的操作。

### D. 功能开关（画面转换）

这个开关将显示跳转到 B8991 号画面



例子使用的画面编号是 B8991 到 8993。在您的系统设计中请根据您的设计计划，改变这些画面编号。(→改变画面编号的方法)

### 3. 地址和 D 脚本说明

示例的运行是通过 GP-PRO/PB III D 脚本功能完成的。示例里使用的 D 脚本说明在下面。（参考：建立和编辑 D 脚本）此外，那些已经使用的 LS 区地址说明也在下面的表格中。（参考：关于 LS 区）

根据您的系统设计需要，去改变那些地址。（参考：更改 D 脚本中使用的地址）

使用的地址	详细说明
“位”地址	——
LS602900	默认设置触发
LS604000	数据收集启动触发
LS604001	写操作触发
LS605000	向上移动7行触发
LS605001	向上移动1行触发
LS605002	向下移动1行触发
LS605003	向下移动7行触发
LS203205	K-tag 数据记录显示的专用触发位继电器，通常为OFF状态。
“字”地址	——
LS2	系统区“年”
LS3	系统区“月”
LS4	系统区“日”
LS5	系统区“小时/分钟”
LS6030	数据收集次数计数
LS6100	数据记录显示, K-tag的间接设置“字”地址。
LS6400-LS6443	数据记录最新数据的创建地址
LS6200-LS6203	字符数据（7个字节）
LS6300-LS6301	32“位”数据（8位十进制数）
LS6302-LS6303	32“位”数据（8位十进制数）
LS6600-LS6626	转换32“位”Dec数据1到ASCII码
LS6700-LS6726	转换32“位”Dec数据2到ASCII码
LS6500-LS6506	CSV文件名
LS7000-LS7017	项目名
LS7022-LS8143	数据收集功能存储的数据显示
临时地址	——
t0080-t0084	地址偏移值放置和循环次数设置地址

\*临时地址只能在 D 脚本内使用。

## D脚本程序说明

## B8991: D脚本

触发：时间触发功能 (1) 在 GP 上电运行后，每秒执行一次。

内容：

```

if([b:LS002900]==0)  如果默认设置“位” OFF
{
    默认数据记录文件名和项目名设置
[w:LS6500]=0x5A4C  “ZL”
[w:LS6501]=0x3030  “00”
[w:LS6502]=0x3030  “00”
[w:LS6503]=0x302E  “0.”
[w:LS6504]=0x4353  “CS”
[w:LS6505]=0x5600  “V”Null
[w:LS6506]=0x0000  Null
[w:LS7000]=0x2020  “Space”
[w:LS7001]=0x4441  “DA”
[w:LS7002]=0x5900  “Y”
[w:LS7004]=0x2C    “,”
[w:LS7005]=0x5449  “TI”
[w:LS7006]=0x4D45  “ME”
[w:LS7007]=0x2C    “,”
[w:LS7008]=0x5445  “TE”
[w:LS7009]=0x5854  “XT”
[w:LS7010]=0x20    “Space”
[w:LS7011]=0x2C    “,”
[w:LS7012]=0x4441  “DA”
[w:LS7013]=0x5441  “TA”
[w:LS7014]=0x3120  “1Space”
[w:LS7015]=0x20    “Space”
[w:LS7016]=0x2C    “,”
[w:LS7017]=0x4441  “DA”
[w:LS7018]=0x5441  “TA”
[w:LS7019]=0x3220  “2”
[w:LS7020]=0x2C    “,”
[w:LS7021]=0xD0A   “CRLF”
[b:LS602900]=1     默认设置“位” ON
}
endif

```

**B8992: D 脚本**

触发：置位（b: LS605000） 每按一次向上移动 7 行的按钮

内容：

```
[w:LS6100]=[w:LS6100]-154      //K-tag 间接字设置地址减去 154
if ([w:LS6100]<=0)              //当结果小于等于 0 时
{
[w:LS6100]=0                    //设置为 0
}
endif
```

触发：置位（b: LS605001） 每按一次向上移动 1 行的按钮

内容：

```
[w:LS6100]=[w:LS6100]-22      //K-tag 间接字设置地址减去 22
if ([w:LS6100]<=0)              //当结果小于等于 0 时
{
[w:LS6100]=0                    //设置为 0
}
Endif
```

触发：置位（b: LS605002） 每按一次向下移动 1 行的按钮

内容：

```
[w:LS6100]=[w:LS6100]+22      //K-tag 间接字设置地址加上 22
if ([w:LS6100]<=3946)          //当结果大于等于 3946 时
{
[w:LS6100]=3946                //设置为 3946
}
endif
```

触发：置位（b: LS605003） 每按一次向下移动 7 行的按钮

内容：

```
[w:LS6100]=[w:LS6100]+154     //K-tag 间接字设置地址加上 154
if ([w:LS6100]>=3946)          //当结果大于等于 3946 时
{
[w:LS6100]=3946                //设置为 3946
}
Endif
```

全

触发：时间触发功能(1) 数据记录开始按钮

内容：

```

if([b:LS604000]==1)           //当数据记录触发“位”为 ON 时
{
[w:LS6030]=[w:LS6030]+1      //日志计数器加 1
Call Data_32bit              //32 “位” Dec-ASCII 转换
Call DATA                   //创建新的日志数据
Call Log                     //数据移动同时数据记录
if ([w:LS6030]==50)         //当日志计数器等于 50 时
{
Call CF_Write               //写入 CF 卡
[w:LS6030]=0               //日志计数器清除为 0
}
endif
[b:LS604000]=0              //数据记录触发“位”复位为 0
}
endif

```

触发：置位（b: LS604001）CF 卡写入按钮

内容：

```

Call CF_Write                //写入 CF 卡
[w:LS6030]=0                //日志计数器清除为 0
[b:LS604001]=0              //CF 卡写入触发“位”复“位”为 0

```

功能：temp\_clr 临时地址清除

内容：

```

[t:0080]=0
[t:0081]=0
[t:0082]=0
[t:0083]=0
[t:0084]=0

```

## D脚本功能

功能: CF\_Write    CF 卡写入

内容:

```

Call temp_clr                               //临时地址清除
if ([w:LS6030]<=25)                          //当日志计数器小于等于 25 时
{
[t:0080]=[w:LS6030]*44+44                   //前面的写入字节计算
_CF_write("\log", [w:LS6500], [w:LS7000], 0, [t:0080], 0)
                                              //前面的数据写入 CF 卡
}
endif
if([w:LS6030]>25)                            //当日志计数器大于 25 时
{
[t:0080]=([w:LS6030]-25)*44+44             //后面的写入字节计算
_CF_write("\log", [w:LS6500], [w:LS7000], 0, 1100, 0)
                                              //前面的数据写入 CF 卡
_CF_write("\log", [w:LS6500], [w:LS7550], 0, [t:0080], 1)
                                              //后面的数据写入 CF 卡
}
endif

```

功能: Log                    数据移动和数据数据记录

内容:

```

memcpy([w:LS7644],[w:LS7622],500)          //向下移动后面的数据
memcpy([w:LS7044],[w:LS7022],600)         //向下移动前面的数据
                                              //最后记录的数据在顶部
Call temp_clr                               //临时地址清除
[t:0080]=22                                 //日志数据第 1 行的字编号
loop([t:0080])
{
                                              //2 个字组合复制到日志地址
    [w:LS7022]#[t:0082]=[w:LS6400]#[t:0081]<<8+[w:LS6401]#[t:0081]
    [t:0081]=[t:0081]+2                     //变量设置
    [t:0082]=[t:0082]+1                     //变量设置
}
endloop

```

## D 脚本功能

功能: 32-bit Dec ..... 32-bit Dec-ASCII 数据转换 .....

内容:

```
[w:LS6600]=[w:LS6300]/10000000)+0x30 //第 8 位数 ASCII 转换
[w:LS6602]=[w:LS6300]%10000000 //第 8 位数余数计算
[w:LS6604]=[w:LS6602]/1000000)+0x30 //第 7 位数 ASCII 转换
[w:LS6606]=[w:LS6602]%1000000 //第 7 位数余数计算
[w:LS6608]=[w:LS6606]/100000)+0x30 //第 6 位数 ASCII 转换
[w:LS6610]=[w:LS6606]%100000 //第 6 位数余数计算
[w:LS6612]=[w:LS6610]/10000)+0x30 //第 5 位数 ASCII 转换
[w:LS6614]=[w:LS6610]%10000 //第 5 位数余数计算
[w:LS6616]=[w:LS6614]/1000)+0x30 //第 4 位数 ASCII 转换
[w:LS6618]=[w:LS6614]%1000 //第 4 位数余数计算
[w:LS6620]=[w:LS6618]/100)+0x30 //第 3 位数 ASCII 转换
[w:LS6622]=[w:LS6618]%100 //第 3 位数余数计算
[w:LS6624]=[w:LS6622]/10)+0x30 //第 2 位数 ASCII 转换
[w:LS6626]=[w:LS6622]%10)+0x30 //第 1 位数 ASCII 转换
[w:LS6700]=[w:LS6302]/10000000)+0x30 //第 8 位数 ASCII 转换
[w:LS6702]=[w:LS6302]%10000000 //第 8 位数余数计算
[w:LS6704]=[w:LS6702]/1000000)+0x30 //第 7 位数 ASCII 转换
[w:LS6706]=[w:LS6702]%1000000 //第 7 位数余数计算
[w:LS6708]=[w:LS6706]/100000)+0x30 //第 6 位数 ASCII 转换
[w:LS6710]=[w:LS6706]%100000 //第 6 位数余数计算
[w:LS6712]=[w:LS6710]/10000)+0x30 //第 5 位数 ASCII 转换
[w:LS6714]=[w:LS6710]%10000 //第 5 位数余数计算
[w:LS6716]=[w:LS6714]/1000)+0x30 //第 4 位数 ASCII 转换
[w:LS6718]=[w:LS6714]%1000 //第 4 位数余数计算
[w:LS6720]=[w:LS6718]/100)+0x30 //第 3 位数 ASCII 转换
[w:LS6722]=[w:LS6718]%100 //第 3 位数余数计算
[w:LS6724]=[w:LS6722]/10)+0x30 //第 2 位数 ASCII 转换
[w:LS6726]=[w:LS6722]%10)+0x30 //第 1 位数 ASCII 转换
```

## D 脚本功能

功能： DATA          创建数据记录的数据

内容：

```
[w:LS6400]=[w:LS0002]>>4)+0x30           // YEAR
[w:LS6401]=[w:LS0002]&0xF)+0x30           // YEAR
[w:LS6402]=0x2F                           // /
[w:LS6403]=[w:LS0003]>>4)+0x30           // MONTH
[w:LS6404]=[w:LS0003]&0xF)+0x30           // MONTH
[w:LS6405]=0x2F                           // /
[w:LS6406]=[w:LS0004]>>4)+0x30           // DAY
[w:LS6407]=[w:LS0004]&0xF)+0x30           // DAY
[w:LS6408]=0x00                           // Null
[w:LS6409]=0x2C                           // ,
[w:LS6410]=[w:LS0005]>>12)+0x30           // TIME
[w:LS6411]=([w:LS0005]>>8)&0xF)+0x30       // TIME
[w:LS6412]=0x3A                           // :
[w:LS6413]=([w:LS0005]>>4)&0xF)+0x30       // MINUTE
[w:LS6414]=[w:LS0005]&0xF)+0x30           // MINUTE
[w:LS6415]=0x2C                           // ,
[w:LS6416]=[w:LS6200]>>8                   // TEXT
[w:LS6417]=[w:LS6200]&0xFF                 // TEXT
[w:LS6418]=[w:LS6201]>>8                   // TEXT
[w:LS6419]=[w:LS6201]&0xFF                 // TEXT
[w:LS6420]=[w:LS6202]>>8                   // TEXT
[w:LS6421]=[w:LS6202]&0xFF                 // TEXT
[w:LS6422]=[w:LS6203]>>8                   // TEXT
[w:LS6423]=0x2C                           // ,
[w:LS6424]=[w:LS6600]                       // 32
[w:LS6425]=[w:LS6604]                       // bit
[w:LS6426]=[w:LS6608]                       // Data
[w:LS6427]=[w:LS6612]                       // part
[w:LS6428]=[w:LS6616]                       // 32
[w:LS6429]=[w:LS6620]                       // bit
[w:LS6430]=[w:LS6624]                       // Data
[w:LS6431]=[w:LS6626]                       // part
[w:LS6432]=0x00                           // Null
[w:LS6433]=0x2C                           // ,
[w:LS6434]=[w:LS6700]                       // 32
[w:LS6435]=[w:LS6704]                       // bit
[w:LS6436]=[w:LS6708]                       // Data
[w:LS6437]=[w:LS6712]                       // part
[w:LS6438]=[w:LS6716]                       // 32
[w:LS6439]=[w:LS6720]                       // bit
[w:LS6440]=[w:LS6724]                       // Data
[w:LS6441]=[w:LS6726]                       // part
[w:LS6442]=0xD                             // CR
[w:LS6443]=0xA                             // LF
```

#### 4. GP、PLC 和画面编辑软件版本

下面的信息对示例中使用的 GP、PLC 和画面编辑软件的情况做了说明。如果您使用了和示例不同的 GP、PLC 等设备，这时必须根据您的系统环境对画面位置和设置进行调整。

(6. 注意事项)

使用的 GP 和 PLC

GP: GP2301L

PLC: Modicon Modbus (SLAVE), Modicon 公司

(协议: Modicon Modbus (SLAVE))

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

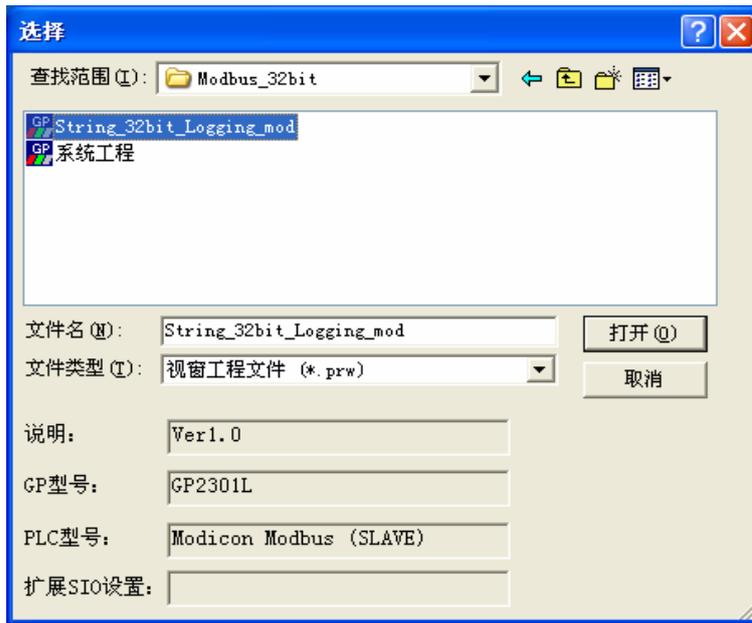
#### 5. 画面复制

如果您的工程文件已创建好，可以直接复制本例到您的工程中。请注意在复制时，地址和画面号不能与工程中已有的地址和画面号重复。(→**确认地址**)

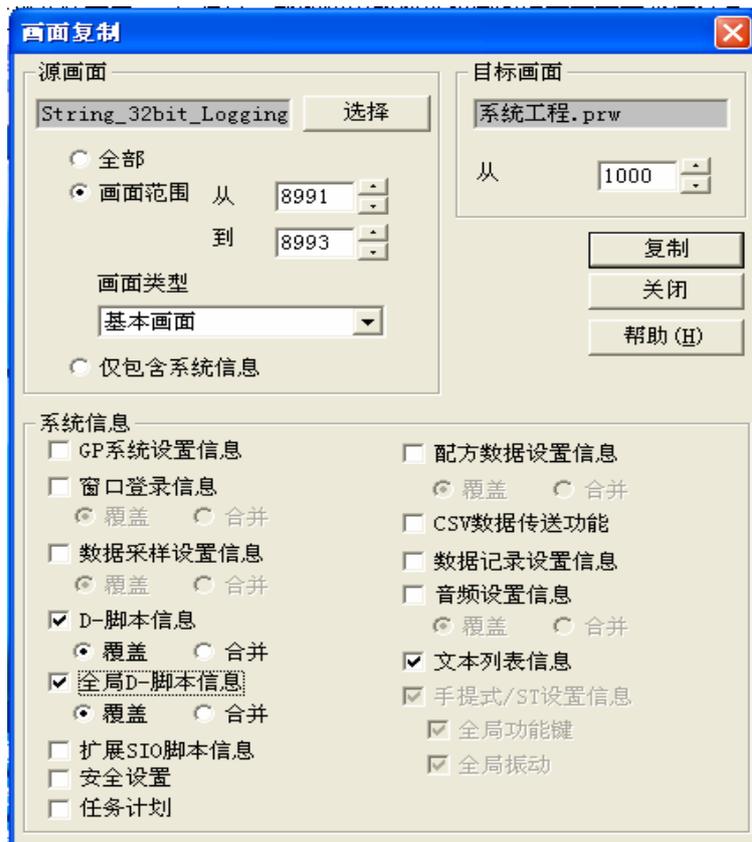
如果您使用的触摸屏型号与本例不同，需要对画面位置和设置进行适当的调整。



打开您的工程文件。在工程管理器窗口的菜单栏里，单击[应用]，选择[画面复制]。



选择示例程序，单击[打开]按钮。



画面范围设置为从 8991 到 8993，画面类型选择基本画面，将复制到您的工程文件里的画面编号指定为 1000。此外，点选 D 脚本信息中的覆盖和全局 D 脚本信息中的覆盖项目。完成设置后，单击[复制]按钮，设置的项目文件将复制到您的工程文件里。

## 6. 注意事项

在使用了不同规格的 GP 时(画面尺寸大小不同), 必须进行画面位置的调整。本示例可以用于 GP2000 系列和 GLC2000 系列。可使用 V6.1 或更高的软件版本。

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。

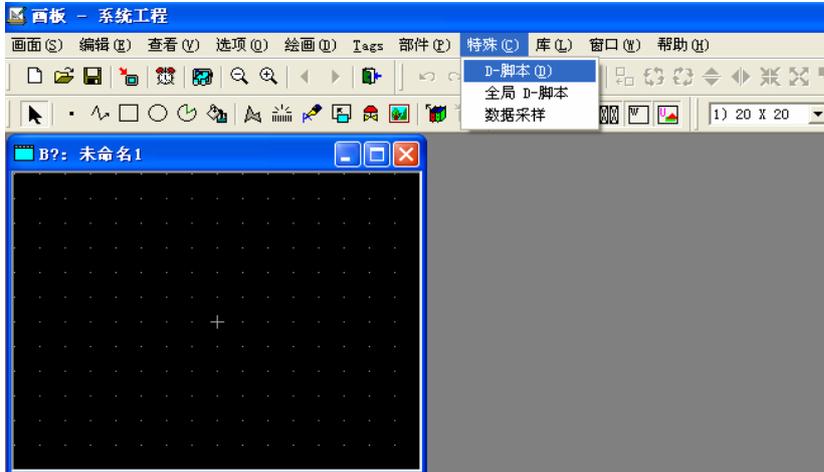
示例: 使用 16 位二进制负数数据



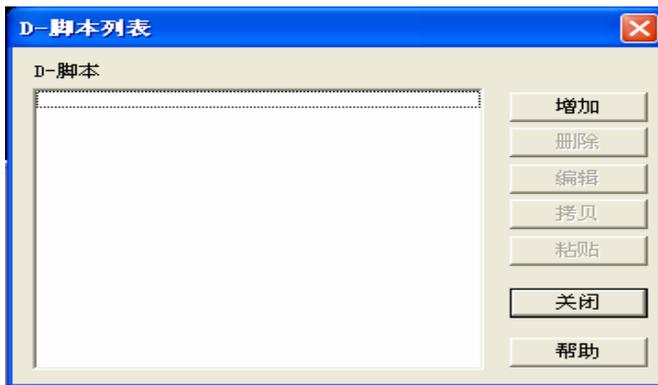
〈附录〉

建立和编辑 D 脚本

建立新的 D 脚本



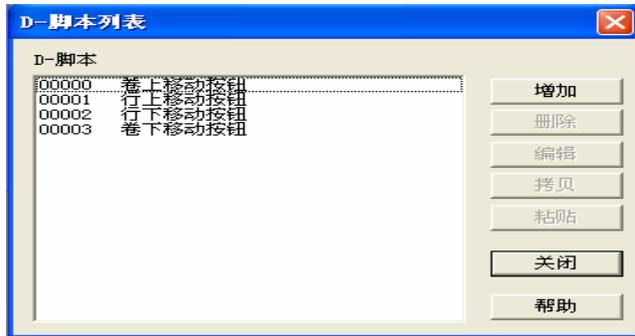
在画面编辑屏幕的菜单栏里，点击[特殊]下拉菜单中的[D 脚本]



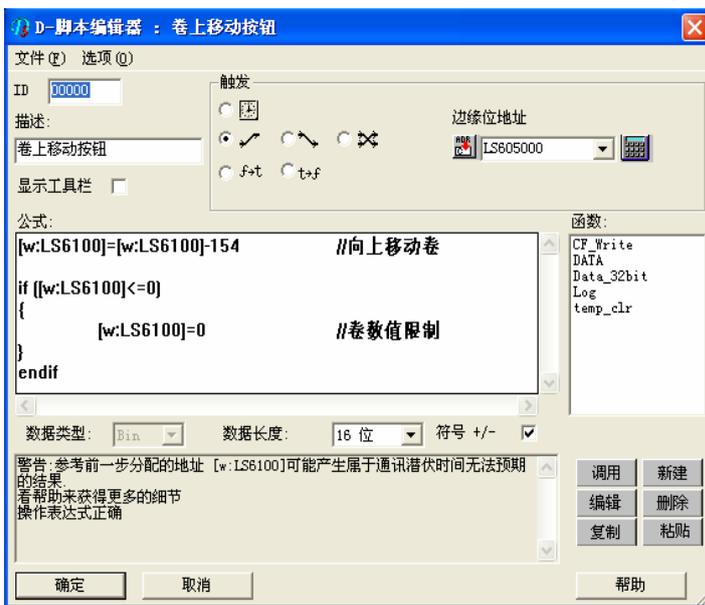
在 D 脚本列表窗口单击 [增加] 按钮。弹出 D 脚本编辑器窗口，使用 D 脚本工具箱输入地址和执行算法，完成编辑输入后单击 [确定]，新编辑的 D 脚本被保存和注册。



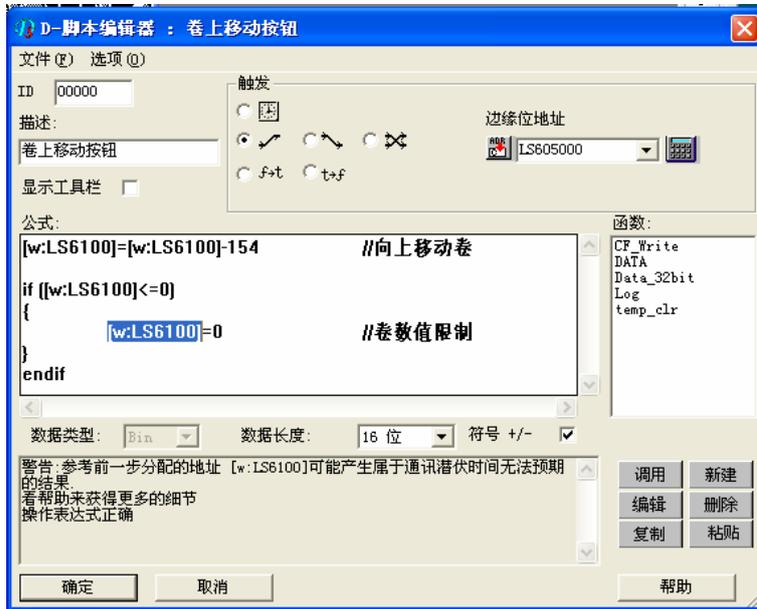
## 编辑D脚本



D 脚本已经注册到 D 脚本列表内。您可以增加新的 D 脚本项目，通过单击[增加]按钮进行。如果需要编辑列表中的 D 脚本，选择希望编辑的 D 脚本，单击[编辑]按钮或在 D 脚本上双击左键。



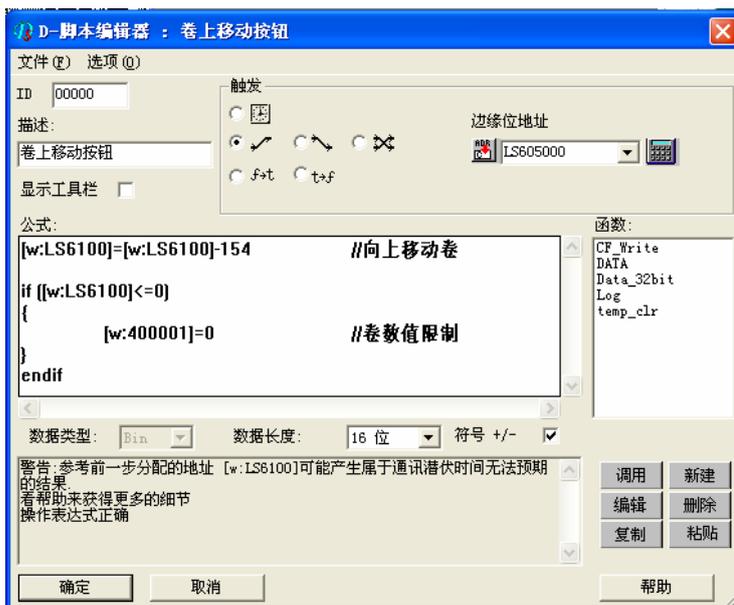
所选择的 D 脚本编辑窗口出现。完成编辑后，单击[确定]按钮保存编辑的内容。



进入希望修改的 D 脚本编辑窗口，在准备修改的地址上双击鼠标左键，这里我们将地址 LS6100 改为 PLC 的地址



输入 400001，单击[确定] 按钮确认输入。



地址改变后的样子

**关于 LS 区**

LS 区是 GP 内部的一个区域，它的分配格式如下表。



在 LS 区里,用户区是在 GP 内部的一个存储器区域，LS 区中的一部分只能用于 GP 内部的过程处理。(→查阅 1.1.2 连接的工具手册)

在 GP2000 和 GLC2000 系列中，LS 区最大可以到 LS8191。

### 确认地址

如何确认当前工程中哪些地址已被画面、D 脚本等使用？下面将详细说明如何确认已使用过的地址。画面号也可通过同样的方法进行确定。

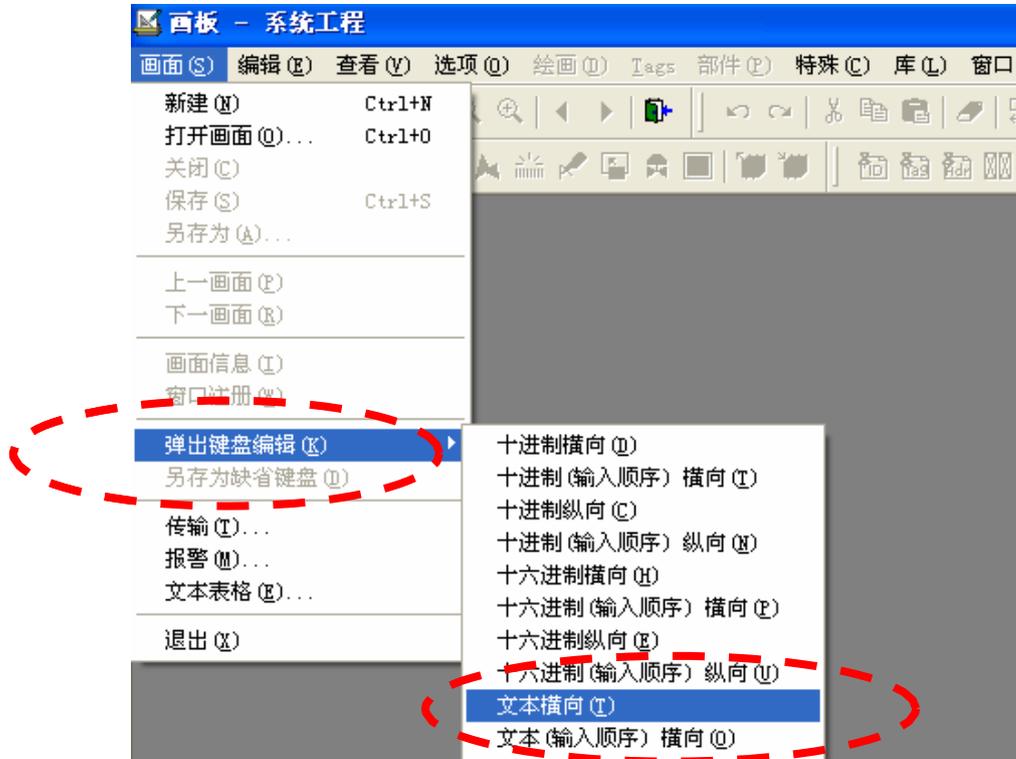


在工程管理器的[应用]菜单中，选择[全局交叉参考]-[列表]命令。



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。

键盘输入显示部件设定的弹出键盘通常其外观尺寸较大。如果一个字符键盘被放在小尺寸的触摸屏内，它将超出其显示范围。为了在小尺寸触摸屏内正常显示字符键盘，就必须去对它的尺寸进行修改。



单击“文本横向”（T），于是弹出键盘库显示在窗口中。选择外观较小的字符串键盘，显示一个（弹出键盘）设计窗口。同时在画面编辑窗口的左下角显示一个“清除区域”按钮和一个“重置区域”按钮。



\*单击“清除区域”同时用鼠标指针选择要显示的范围可以调整其显示在画面里的内容和大小（根据需要调整键盘外观和键的内容）