

SC165C010 软件功能手册

(2004-11-09 版本: 1.8.3 杭州士兰微电子股份有限公司)

内容提要:

- ☒ 简介
- ☒ 数据格式
- ☒ 软件功能详解
- ☒ 对象类型汇总
- ☒ 对象操作方法汇总
- ☒ 过滤器/填充模板
- ☒ 功能索引

一、简介

液晶模块的功能主要是由液晶显示控制器的软件功能决定。通用图形点阵液晶显示控制器的软件功能有以下几个特点:

- ◆ 最多可支持 16 种自定义字库。
- ◆ 具有字符、字符串和位图显示功能及填充功能。
- ◆ 具有灵活的页面操作功能。
- ◆ 多种逻辑显示方式: 普通、复制、与、或、非、异或。
- ◆ 支持点阵反显、下划线。
- ◆ 256 种填充模板/过滤器。
- ◆ 具有绘制点、水平线、垂直线、矩形框、带阴影矩形框和任意两点间线段的功能。
- ◆ 支持自定义线型。
- ◆ 任意宽度和高度(最大为 24×24) 的光标显示。
- ◆ 引入活动区域和可视活动区域的概念, 超出可视活动区域的显示内容被自动裁减。
- ◆ 支持安全的内存操作, 可在 RAM 和 E²PROM 中开辟一块区域作为一般数据存区取用, 这在一些场合下可以减少外部存储器件。
- ◆ 支持复杂的显示对象, 功能更强大, 操作更方便。
- ◆ 支持点阵显示信息外灌, 保持可扩展性。

二、数据格式

使用通用图形点阵液晶显示控制器的主要手段就是向其发送参数和命令。每个功能命令所需的参数个数并不一定相同, 但是 MCU 向液晶控制器传送数据的方式是相同的, 即: 先按顺序向液晶控制器的数据通道连续发送参数, 再向命令通道发送命令代码。如果该命令有返回参数, 则在命令执行完毕, 液晶空闲时从数据通道依次读出这些参数。

假设对 MCS51 单片机而言, 液晶控制器的数据通道地址为 LCD_DA, 命令通道地址为 LCD_CA (这是由具体的硬件电路连接决定的), 则以下一段程序实现对液晶模块的页面信息设置:

```
LCALL  WAIT_LCD_IDLE      ;等待液晶空闲
MOV    DPTR,  #LCD_DA     ;液晶的数据通道地址
MOV    A,      #01H       ;[0]<6..4> = 0, [0]<2..0> = 1
MOVX   @DPTR,  A           ;第 0 页为显示页面, 第 1 页为操作页面
MOV    DPTR,  #LCD_CA     ;液晶的命令通道地址
MOV    A,      #F_SETPAGEINFO
MOVX   @DPTR,  A
```

我们再来看一段多字节参数的命令的使用方法，以下一段程序要求液晶显示一条斜线段（起点坐标为(0, 0)，终点坐标为(32, 48)）：

```
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
MOV      A,       #00H       ;[0]
MOVX     @DPTR,    A
MOV      A,       #00H       ;[1]
MOVX     @DPTR,    A
MOV      A,       #00H       ;[2]
MOVX     @DPTR,    A
MOV      A,       #00H       ;[3]
MOVX     @DPTR,    A
MOV      A,       #00H       ;[4]
MOVX     @DPTR,    A
MOV      A,       #20H       ;[5]
MOVX     @DPTR,    A
MOV      A,       #00H       ;[6]
MOVX     @DPTR,    A
MOV      A,       #30H       ;[7]
MOVX     @DPTR,    A
MOV      A,       #FFH       ;[8]
MOVX     @DPTR,    A
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_DRAWLINE
MOVX     @DPTR,    A
```

下面这段程序有 1 个返回参数，它的功能是获得点阵显示的过滤器编号，没有输入参数：

```
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_GETFILTER
MOVX     @DPTR,    A
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
MOV      A,       @DPTR      ;A = 过滤器编号
```

等待液晶空闲的子程序 WAIT_LCD_IDLE 代码如下：

```
WAIT_LCD_IDLE:              ;等待液晶空闲
    MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
WAIT_LCD_LP:
    MOV      A,       @DPTR      ;获取液晶忙状态字, 它从命令通道读出
    CJNE     A, #0FFH, WAIT_LCD_LP ;忙, 循环等待
    RET
```

液晶处于忙状态时无法接收参数和命令，因此，等待液晶空闲的子程序是很常用的。

三、软件功能详解

以下逐条介绍通用图形点阵液晶显示控制器的所有软件功能命令。但在正式介绍之前需对一些符号和颜色的含义作些解释，以便于阅读。

1. 符号

- ◆ N(M)

出现在“输入参数个数”和“输出参数个数”栏中，N表示参数的个数，括号内的M表示字节数。

例如“6(8)”：表示总共用8个字节表示6个参数。

- ◆ [I], [I1`I2]

字节符号。出现在“输入参数说明”、“输出参数说明”、“使用说明”和“使用实例”栏中，I、I1、I2均表示参数字节的传送次序，[I]表示第I个字节（它可能构成8位的参数，也可能只是某个参数的其中一个字节），[I1`I2]表示第I1个字节和第I2个字节构成的16位数（其中[I1]为高字节，[I2]为低字节）。

例如“[0]”：表示第0个字节。

又如“[2`3]”：表示[2]为高字节，[3]为低字节，它们构成一个16位数。

- ◆ , <B2..B1>

位符号。出现在“输入参数说明”、“输出参数说明”、“使用说明”和“使用实例”栏中，B、B1、B2均表示位序号，表示某个字节的第B位，<B2..B1>表示某个字节的第B2位至B1位，它们常与字节符号一起构成具体的位。

比如“[0]<3>”：表示第0个字节的第3位。

又如“[0]<6..4>”：表示第0个字节的第6到4位。

- ◆ LCD_DA

这在前文已经出现过，它表示液晶的数据通道地址，由具体的硬件连接决定。

- ◆ LCD_CA

表示液晶的命令通道地址，与LCD_DA一样，它也是由具体的硬件连接决定。

- ◆ WAIT_LCD_IDLE

等待液晶空闲的子程序，在前文中提到过，并有MCS51程序的例子。

2. 颜色

- ◆ 深蓝 - 功能名称
- ◆ 红色 - 功能命令代码
- ◆ 青色 - 说明性文字
- ◆ 深红 - 符号和编号
- ◆ 深黄 - 参数描述
- ◆ 蓝色 - 特别应注意的说明性文字

功能名称:	F_SETBASIC	功能命令代码:	00H
功能简介:	设置液晶的基本参数		
输入参数个数:	6(8)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 页面文本宽度	8 位无符号	1, 40	文本像素
[1] 页面图形高度	8 位无符号	1, 240	图形像素
[2] 随机存储器(RAM)容量-1 (如 31:表示值为 32×256 字节=8K 字节)	8 位无符号	3, 255	256 字节
[3] 页面数目	8 位无符号	1, 8	-
[4`5] 通用显示信息缓冲区大小	16 位无符号	0, 65535	字节
[6`7] 自由存取区大小	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> 如果液晶模块中有可编程只读存储器(E²PROM), 并且 E²PROM 被预置过, 则该功能失效, 执行将它不产生任何效果。 输入参数必需符合液晶模块的实际参数, 否则执行其它功能命令将可能产生意想不到的错误. 无论如何, 应该遵循这样的关系式: $[0] \times [1] \times [3] + [4`5] + [6`7] + 256 \leq ([2] + 1) \times 256$① 一般情况下, 该功能命令为液晶模块上电复位后第一个需要执行的命令, 且不再执行第二遍. 因为一旦执行了该功能命令, 随机存储器的空间将被重新分配, 相应的数据指针也将被复位, 这样可能导致数据丢失。 			
使用实例:			
<ul style="list-style-type: none"> 对于一块 240×64 点阵的液晶模块, 其 RAM 容量为 32K 字节, 没有 E²PROM, 其各基本参数的值按顺序可以设置如下: [0]1EH, [1]40H, [2]7FH, [3]08H, [4]20H, [5]00H, [6]23H, [7]00H. <p>用 MCS51 程序写成:</p> <pre> LCALL WAIT_LCD_IDLE ;等待液晶空闲 MOV DPTR, #LCD_DA ;液晶的数据通道地址 MOV A, #1EH ;[0]页面文本宽度为 30 MOVX @DPTR, A MOV A, #40H ;[1]页面图形高度为 64 MOVX @DPTR, A MOV A, #7FH ;[2]随机存储器容量为 32K 字节 MOVX @DPTR, A MOV A, #08H ;[3]页面数目为 8 MOVX @DPTR, A MOV A, #20H ;[4`5]通用显示信息缓冲区大小为 8K 字节 MOVX @DPTR, A MOV A, #00H MOVX @DPTR, A MOV A, #23H ;[6`7]自由存取区大小为 8960 字节 MOVX @DPTR, A MOV A, #00H MOVX @DPTR, A </pre>			

```
MOV    DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV    A,        #F_SETBASIC
MOVX   @DPTR,    A
;...
```

其中, [0], [1], [2] 三个参数根据实际情况设置, [3], [4`5], [6`7] 可以在遵循关系式❶的前提下根据需要进行设置.

本例中: 页面数目为 8, 通用显示信息缓冲区大小为 2000H(8K) 字节, 自由存取区大小为 2300H(8960) 字节.

- ◆ 对于一块 320×240 点阵的液晶模块, 其 RAM 容量为 32K 字节, 没有 E²PROM, 其各基本参数的值按顺序可以设置如下: [0]28H, [1]F0H, [2]7FH, [3]03H, [4]08H, [5]00H, [6]04H, [7]00H.

其中, [3] 的值最多只能设置为 03H, 否则 [0]×[1]×[3] 将大于 ([2] + 1)×256.

本例中: 页面数目为 3, 通用显示信息缓冲区大小为 0800H(2K) 字节, 自由存取区大小为 0400H(1K) 字节. 还有 896 个字节的 RAM 单元没有使用.

- ◆ 对于有 E²PROM 的液晶模块, 则一般情况下不使用该功能命令, 而通用“页面编辑器”设置这些基本参数并预置到 E²PROM 中. 液晶上电复位后会自动从 E²PROM 中获取这些参数, 并使该功能命令失效.

相关功能:

- ◆ F_GETBASIC

功能名称:	F_GETBASIC	功能命令代码:	01H
功能简介:	获取液晶基本参数的设置值		
输入参数个数:	0	输出参数个数:	7(10)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] 页面文本宽度	8 位无符号	1, 40	文本像素
[1] 页面图形高度	8 位无符号	1, 240	图形像素
[2] 随机存储器 (RAM) 容量-1 (如 31: 表示值为 32×256 字节=8K 字节)	8 位无符号	3, 255	256 字节
[3] 页面数目	8 位无符号	1, 8	-
[4`5] 通用显示信息缓冲区大小	16 位无符号	0, 65535	字节
[6`7] 自由存取区大小	16 位无符号	0, 65535	字节
[8`9] 可编程自由存取区大小	16 位无符号	0, 65535	字节
使用说明:	<ul style="list-style-type: none">◆ 该功能命令可以在液晶的任何空闲时执行.◆ 无论 F_SETBASIC 功能是否有效, 该功能总是有效的.		
使用实例:	<ul style="list-style-type: none">◆ 由于液晶基本参数关系到液晶模块的正常运行, 为了保证基本参数设置的正确性, 建议在设置基本参数之后读取它们并做比较, 确保基本参数设置的正确性. 但是应该注意该功能的输出参数个数比 F_SETBASIC 功能的输入参数个数要多.		
相关功能:	◆ F_SETBASIC		

功能名称:	F_SETOSCFREQ	功能命令代码:	02H
功能简介:	设置外接晶振频率		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 液晶显示控制器外接晶振的频率	8 位无符号	20, 160	100KHz
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> 如果液晶模块中没有可编程只读存储器 (E²PROM), 则该功能命令可在液晶基本参数设置 (F_SETBASIC) 之后的任何空闲时执行. 如果液晶模块中有可编程只读存储器 (E²PROM), 并且 E²PROM 被预置过, 则该功能仍然有效, 且可以在任何空闲时执行. 输入参数以 100KHz 为单位, 如果晶振的实际频率不能被 100KHz 整除, 则取大于它的最小值设置即可. 外接晶振频率的设置值将影响控制器内部的定时机制, 比如影响到光标闪烁频率和可编程自由存取区的编程周期, 一般情况下总是设置成与实际值接近. 如果想加快内部定时频率, 可以将该功能命令的输入参数设成比晶振实际频率小的值, 但是这样可能会使可编程自由存取区的写操作出现错误. 			
使用实例:			
<ul style="list-style-type: none"> 如果外接晶振的频率为 10MHz (这最常见), 则参数 [0] 设置为 64H. 这种符合实际的设置下, 光标闪烁的频率为 0.7 秒 (即每隔 0.35 秒光标切换一次显示状态). 如果外接晶振的频率为 8MHz, 而参数 [0] 设置成 28H, 则光标闪烁的频率加快了 1 倍, 为 0.35 秒. 			
相关功能:			
<ul style="list-style-type: none"> F_GETOSCFREQ F_SETBASIC 			

功能名称:	F_GETOSCFREQ	功能命令代码:	03H
功能简介:	获取外接晶振频率的设置值		
输入参数个数:	0	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] 液晶显示控制器外接晶振的频率	8 位无符号	20, 160	100KHz
使用说明:			
(无)			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> F_SETOSCFREQ 			

功能名称:	F_SETDISP	功能命令代码:	04H
-------	-----------	---------	-----

功能简介：		设置液晶显示开关		
输入参数个数：		1 (1)	输出参数个数：	0
输入参数说明：		参数编码方式	值域	单位
[0] 开关状态： 0 表示关显示，非 0 表示开显示		—	—	—
输出参数说明：		参数编码方式	值域	单位
—		—	—	—
使用说明：				
<ul style="list-style-type: none">液晶显示的开关状态只表示液晶模块上显示驱动电路的工作与否，并不影响任何其它操作。如果执行该功能命令关闭液晶显示，则液晶屏上的显示信息消失，但显示数据依然可以送往液晶模块的随机存储器中，一旦开启液晶显示，这些信息便重新显示在液晶屏上。该功能命令是否有效还要视具体的液晶模块而定。有些液晶模块放弃使用该功能的控制引脚，从而使该功能失效。获取液晶显示开关状态请执行 F_GETLSW 功能命令。				
使用实例：				
(略)				
相关功能：				
<ul style="list-style-type: none">F_GETLSW				

功能名称:	F_GETLSW	功能命令代码:	05H
功能简介:	获取液晶状态字		
输入参数个数:	0	输出参数个数:	6(1)
输入参数说明:	参数编码方式	值域	单位
—	—	—	—
输出参数说明:	参数编码方式	值域	单位
E ² PROM 预置标志位			
[0]<0>	1:E ² PROM 存在且被预置过, 0:E ² PROM 不存在或者没有被预置过	1 位	0, 1
可编程自由存取区写保护标志位			
[0]<1>	1: 允许写, 0: 写保护	1 位	0, 1
中英文全字库 ROM 存在标志位			
[0]<2>	1: 存在, 0: 不存在	1 位	0, 1
[0]<4>	1: 液晶显示开, 0: 液晶显示关	1 位	0, 1
[0]<5>	1: 光标显示开, 0: 光标显示关	1 位	0, 1
[0]<6>	1: 光标闪烁开, 0: 光标闪烁关	1 位	0, 1
使用说明:			
◆ 输出参数的第 0 位和第 2 位显示的状态由液晶上电复位时自动设置, 因此没有相应的设置它们的功能命令。			
◆ 输出参数的第 3 位和第 7 位保留。			
◆ 设置可编程自由存取区写保护标志位([0]<1>)的功能命令为:F_EEPWREN。			
◆ 设置液晶显示开关([0]<4>)的功能命令为:F_SETDISP。			
◆ 设置光标显示开关([0]<5>)和光标闪烁开关([0]<6>)的功能命令分别为:F_SETCURSOR 和			

F_SETCURGLIT.
使用实例:
(略)
相关功能:
◆ F_EEPWREN
◆ F_SETDISP
◆ F_SETCURSOR
◆ F_SETCURGLIT

功能名称:	F_SETPAGEINFO	功能命令代码:	08H
功能简介:	设置页面信息, 包括显示页号和操作页号		
输入参数个数:	2(1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] <6..4>	将该序号的页面设置成显示页面	3 位无符号	0, 7
[0] <2..0>	将该序号的页面设置成操作页面	3 位无符号	0, 7
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ 该功能命令虽然有两个输入参数, 但只需一个字节即可表示, 这个字节的第 6 到 4 位用来表示显示页面的序号, 第 2 到 0 位表示操作页面的序号. 第 3 位和第 7 位是无效的.			
◆ 输入参数的值大于或等于页面数则操作失败, 不产生任何效果. 页面数可以通过 F_GETBASIC 功能获得.			
◆ 如果操作页号发生改变, 则光标显示将被关闭.			
使用实例:			
◆ 如果页面数目已经设置成 4, 现在需要将第 1 页设置成显示页面, 第 3 页设置成操作页面, 则可将参数 [0] 设置成 13H(X001_X011B).			
这样设置之后, 液晶屏将显示第 1 页的内容, 并且此后所有的显示操作都将针对第 3 页进行.			
◆ 一般情况下, 我们都把显示页面和操作页面设置成同一个页面, 但是, 如果显示过程耗费的时间较长的话, 效果就不是很好. 利用该功能可以先将操作页面与显示页面分开, 执行显示命令之后, 再将操作页面所在的页面设置成显示页面. 如以下一些 MCS51 的代码:			
LCALL WAIT_LCD_IDLE ;等待液晶空闲			
MOV DPTR, #LCD_DA ;液晶的数据通道地址			
MOV A, #01H ;[0]<6..4> = 0, [0]<2..0> = 1			
MOVX @DPTR, A ;第 0 页为显示页面, 第 1 页为操作页面			
MOV DPTR, #LCD_CA ;液晶的命令通道地址			
MOV A, #F_SETPAGEINFO			
MOVX @DPTR, A			
;... ;显示操作, 如显示字符, 位图, 绘图等, 此时屏上			
; ;并无这些显示结果			
MOV DPTR, #LCD_DA ;液晶的数据通道地址			
MOV A, #11H			

MOVX	@DPTR, A	;第 1 页为显示页面, 第 1 页为操作页面
MOV	DPTR, #LCD_CA	;液晶的命令通道地址
MOV	A, #F_SETPAGEINFO	
MOVX	@DPTR, A	;这条命令之后, 屏上显示出刚才操作的结果
LCALL	WAIT_LCD_IDLE	;等待液晶空闲
实现了消除显示过渡过程的功能.		
相关功能:		
◆ F_GETPAGEINFO		
◆ F_SETBASIC		
◆ F_GETBASIC		
◆ F_GETLSW		
◆ F_SETCURSOR		

功能名称:	F_GETPAGEINFO	功能命令代码:	09H
功能简介:	获取页面信息, 包括显示页号和操作页号		
输入参数个数:	0	输出参数个数:	2(1)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] <6..4>	该序号的页面为显示页面	3 位无符号	0, 7
[0] <2..0>	该序号的页面为操作页面	3 位无符号	0, 7
使用说明:			
◆ 该功能命令虽然有两个输出参数, 但只用一个字节表示, 这个字节的第 6 到 4 位用来表示显示页面的序号, 第 2 到 0 位表示操作页面的序号.			
使用实例:			
◆ 请看以下 MCS51 程序:			
LCALL WAIT_LCD_IDLE ;等待液晶空闲			
MOV DPTR, #LCD_CA ;液晶的命令通道地址			
MOV A, #F_GETPAGEINFO			
MOVX @DPTR, A ;获取页面信息			
LCALL WAIT_LCD_IDLE ;等待液晶空闲			
MOV DPTR, #LCD_DA			
MOVX A, @DPTR ;A<6..4>为显示页号, A<2..0>为操作页号			
;...			
相关功能:			
◆ F_SETPAGEINFO			

功能名称:	F_SETACTRECT	功能命令代码:	0CH
-------	--------------	---------	-----

功能简介：	设置活动区域			
输入参数个数：	4 (5)	输出参数个数：	0	
输入参数说明：		参数编码方式	值域	单位
[0]	活动区域左上角文本横坐标	8 位补码	-128, 127	文本像素
[1`2]	活动区域左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[3]	活动区域的文本宽度	7 位无符号	0, 127	文本像素
<6..0>				
[4]	活动区域的图形高度	8 位无符号	0, 255	图形像素
输出参数说明：		参数编码方式	值域	单位
-	-	-	-	-
使用说明：				
<ul style="list-style-type: none">◆ 活动区域是一个抽象的概念, 我们并不会直观地看到这个区域, 它只是限制显示内容使之不超出这个区域. 重新设置活动区域不会影响已经显示在屏上的内容, 它只会影响设置之后的绝大部分点阵显示, 所有图形显示以及光标显示.◆ 与“活动区域”关系密切的另一个重要概念是“可视活动区域”. 其实, 真正限制显示内容的是可视活动区域. 该区域是活动区域与整个页面区域的重叠部分. 可视活动区域无需单独设置, 它在执行设置活动区域的功能命令时被自动设置.◆ 清除操作页面是唯一不受活动区域限制的点阵显示操作.◆ 活动区域始终是针对操作页面而言的, 因为所有的显示操作均以操作页面为目标.◆ 所有的显示坐标都是相对于活动区域的左上角而言的.◆ 坐标值可以是负数, 它的值以补码形式表示; 宽度和高度始终大于或等于 0.◆ 输入参数应落在其值域内, 对于参数[1`2]而言, 它的值会被自动限制在值域内, 对于参数[3]而言, 它的第 7 位是无效的.				
使用实例：				
<ul style="list-style-type: none">◆ 对于文本宽度为 16, 图形高度为 64 的页面而言, 若输入参数为: [0]FCH, [1]FFH, [2]F8H, [3]18H, [4]40H 则活动区域的左上角坐标为 (-4, -8), 文本宽度为 24, 图形高度为 64. 用图表示为:				
<p>其中粗线框为页面区域, 虚线框为活动区域, 阴影部分为可视活动区域.</p> <p>这样设置之后, 所有关于显示的坐标均以 (-4, 8) 为原点. 但是应该注意, 这里的横坐标为文本横坐标.</p>				
相关功能：				
<ul style="list-style-type: none">◆ F_SETACTFULL◆ F_GETACTRECT◆ F_SCRCLR◆ F_SETBASIC				

功能名称:	F_GETACTRECT	功能命令代码:	ODH
功能简介:	获取活动区域		
输入参数个数:	0	输出参数个数:	8(10)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0]	活动区域左上角文本横坐标	8 位补码	-128, 127 文本像素
[1`2]	活动区域左上角图形纵坐标	16 位补码	-1024, 1023 图形像素
[3]	活动区域的文本宽度	8 位无符号	0, 127 文本像素
[4]	活动区域的图形高度	8 位无符号	0, 255 图形像素
[5]	可视活动区域左上角文本横坐标	8 位无符号	0, 40 文本像素
[6]	可视活动区域左上角图形纵坐标	8 位无符号	0, 240 图形像素
[7]	可视活动区域右下角文本横坐标	8 位补码	-1, 39 文本像素
[8, 9]	可视活动区域右下角图形纵坐标	16 位补码	-1, 239 图形像素
使用说明:			
◆ 读出的可视活动区域若出现右边文本横坐标比左边的小, 则表示可视活动区域为空. 若下边的图形纵坐标比上边的小, 也表示可视活动区域为空.			
使用实例:			
◆ 一般情况下, 我们更关心活动区域左上角的坐标, 因为它是显示坐标的参照点. 因此可以只读出这两个参数而舍弃其它的参数. 下面有一段 MCS51 程序的示例:			
<pre> LCALL WAIT_LCD_IDLE ;等待液晶空闲 MOV DPTR, #LCD_CA ;液晶的命令通道地址 MOV A, #F_GETACTRECT MOVX @DPTR, A ;获取活动区域 LCALL WAIT_LCD_IDLE ;等待液晶空闲 MOV DPTR, #LCD_DA MOVX A, @DPTR MOV R2, A ;R2 = 活动区域左上角文本横坐标 MOVX A, @DPTR MOV R3, A ;R3 = 活动区域左上角图形纵坐标(高字节) MOVX A, @DPTR MOV R4, A ;R4 = 活动区域左上角图形纵坐标(低字节) ;... ;无需读出余下的 6 个参数 </pre>			
相关功能:			
◆ F_SETACTRECT			
◆ F_SETACTFULL			

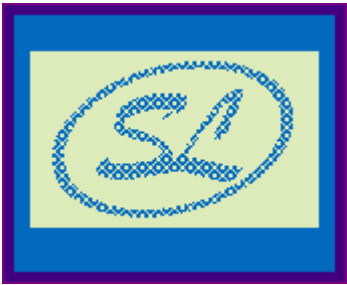
功能名称:	F_SETACTFULL	功能命令代码:	OEH
功能简介:	设置整个页面区域为活动区域		
输入参数个数:	0	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位

输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
(无)			
使用实例:			
(略)			
相关功能:			
◆ F_GETACTRECT			
◆ F_SCRCLR			
◆ F_SETBASIC			

功能名称:	F_SETFILTER	功能命令代码:	10H
功能简介:	设置过滤器		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 过滤器编号	8 位无符号	0, 255	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
◆ 点阵显示的普通方式是指:送往目标显示区域的点阵数据(文本像素)先和过滤器进行与运算,再和反显方式字节进行异或运算,最后才送到指定的坐标处进行显示. 另外,普通方式的点阵显示可能还受下划线方式字节的影响.			
◆ 受下划线方式字节影响的点阵显示,其点阵信息在可视活动区域内的最后一行点阵数据先和下划线方式字节异或,再进行过滤和反显.			
◆ 过滤方式,反显方式和下划线方式的初始值分别为:FFH, 00H, 00H.			
◆ 点阵显示的逻辑方式以及非点阵显示并不受过滤器,反显方式和下划线方式的影响.			
使用实例:			
以下各例均针对下面这幅原始位图:			
			
(原始位图显示在 160×128 的液晶屏上,活动区域为整个页面)			
◆ 如果过滤器,反显方式和下划线方式分别为:FFH, 00H, 00H, 则位图的显示结果与原图相同.			
◆ 如果过滤器,反显方式和下划线方式分别为:DCH, 00H, 00H, 则位图的显示结果如下:			



◆ 如果过滤器, 反显方式和下划线方式分别为:DCH, FFH, 00H, 则位图的显示结果如下:



◆ 如果过滤器, 反显方式和下划线方式分别为:B3H, 00H, FFH, 则位图的显示结果如下:



◆ 如果过滤器, 反显方式和下划线方式仍然分别为:B3H, 00H, FFH, 活动区域的左上角为(0, 0), 但活动区域的文本宽度和图形高度分别改为 16 和 80, 则位图的显示结果如下:



◆ 选择一系列过滤器, 并使用普通方式进行显示, 可以实现位图淡入的效果.

相关功能:	
◆ F_GETFILTER	
◆ F_SETINVERT	
◆ F_SETUNDERLINE	

功能名称:	F_GETFILTER	功能命令代码:	11H
-------	-------------	---------	-----

功能简介:	获取过滤器编号		
输入参数个数:	0	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] 过滤器编号	8 位无符号	0, 255	-
使用说明:	(无)		
使用实例:	(略)		
相关功能:	◆ F_SETFILTER		

功能名称:	F_SETINVERT	功能命令代码:	12H
功能简介:	设置反显方式		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 反显方式字节	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	◆ (见 F_SETFILTER 功能)		
使用实例:	◆ (见 F_SETFILTER 功能)		
相关功能:	◆ F_GETINVERT ◆ F_SETFILTER ◆ F_SETUNDERLINE		

功能名称:	F_GETINVERT	功能命令代码:	13H
功能简介:	获取反显方式		
输入参数个数:	0	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] 反显方式字节	-	-	-
使用说明:	(无)		
使用实例:	(略)		
相关功能:			

◆ F_SETINVERT

功能名称:	F_SETUNDERLINE	功能命令代码:	14H
功能简介:	设置下划线方式		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 下划线方式字节	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	◆ (见 F_SETFILTER 功能)		
使用实例:	◆ (见 F_SETFILTER 功能)		
相关功能:	◆ F_GETUNDERLINE ◆ F_SETFILTER ◆ F_SETINVERT		

功能名称:	F_GETUNDERLINE	功能命令代码:	15H
功能简介:	获取下划线方式		
输入参数个数:	0	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0] 下划线方式字节	-	-	-
使用说明:	(无)		
使用实例:	(略)		
相关功能:	◆ F_SETUNDERLINE		

功能名称:	F_SETCURSOR	功能命令代码:	18H
功能简介:	光标显示开关		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 0:关, 非 0:开	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			

- ◆ 当光标处有一系列独立显示需要连续进行时(比如动画), 最好先关闭光标, 待显示操作全部结束后再开启.
- ◆ 光标不用时, 最好是关闭它, 而不要只把其宽度或高度设为 0 或把其坐标移出页面的有效区域.
- ◆ 液晶上电复位后, 光标显示是关闭的.

使用实例:

- ◆ 光标只在操作页面中存在, 若当前操作页面不是显示页面时, 即使光标开启且在可视活动区内也无法在液晶屏上见到它.




相关功能:

- ◆ F_GETLSW
- ◆ F_SETCURGLIT
- ◆ F_MOVECURSOR
- ◆ F_RESIZECURSOR

功能名称:	F_SETCURGLIT	功能命令代码:	19H
功能简介:	光标闪烁开关		
输入参数个数:	1(1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 0:关, 非 0:开	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	<ul style="list-style-type: none"> ◆ 如果液晶外接晶振频率的设置值与实际值相等, 且光标闪烁开, 则光标以接近 1 秒的周期闪烁, 即每隔 0.5 秒切换一次显示状态. ◆ 液晶上电复位后, 光标闪烁是开启的. 		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_SETOSCFREQ ◆ F_GETLSW ◆ F_SETCURSOR ◆ F_MOVECURSOR ◆ F_RESIZECURSOR 		

功能名称:	F_MOVECURSOR	功能命令代码:	1AH
功能简介:	移动光标		
输入参数个数:	2(4)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 新位置左上角的文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3] 新位置左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
输出参数说明:	参数编码方式	值域	单位
-	-	-	-

使用说明:
<ul style="list-style-type: none">坐标参数会被自动限制的值域内.移动光标是对光标的最常用的操作,从本质上它不改变光标的尺寸,但是它受当前活动区域的限制,超出可视活动区域的部分将被裁减.光标的坐标指其左上角的坐标,并且是以活动区域左上角为参照的.光标属性的设置只受当前可视活动区域的影响,如果设置完成后活动区域发生改变则仍然不影响当前的光标属性和裁减情况,因此,一般的应用总是在活动区域改变前先关闭光标,在需要时再开启,否则有可能出现无法设置新属性的情况.
使用实例:
(略)
相关功能:
<ul style="list-style-type: none">F_SETACTRECTF_SETCURSORF_RESIZECURSOR

功能名称:	F_RESIZECURSOR	功能命令代码:	1BH
功能简介:	改变光标尺寸		
输入参数个数:	2(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 光标的图形宽度	8 位无符号	0, 24	图形像素
[1] 光标的图形高度	8 位无符号	0, 24	图形像素
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:	<ul style="list-style-type: none">(参见 F_MOVECURSOR 的使用说明)尺寸参数会被自动限制在值域内.光标的宽度单位是图形像素,而它的横坐标单位是文本像素,这个现象是比较特别的.光标的宽度尽量是 8 的整倍数,这样它消耗的资源比较小.		
使用实例:	<ul style="list-style-type: none">光标尺寸的灵活性使得我们可以实现横向光标和纵向光标,也可以实现块状光标.比如光标的宽度和高度分别设置为 8 和 1,则它的形状为: 如果光标的宽度和高度分别设置为 2 和 16,则它的形状为: 如果光标的宽度和高度分别设置为 8 和 16,则它的形状为:  <p>在插入编辑方式下,一般我们选用横向光标或纵向光标;在替换编辑方式下,通常选用块状光标,有时也选用横向光标.</p>		
相关功能:	<ul style="list-style-type: none">F_SETACTRECT		

- ◆ F_SETCURSOR
- ◆ F_MOVECURSOR

功能名称:	F_MODISTRCUR	功能命令代码:	1CH
功能简介:	将字符串显示后产生的光标信息赋给光标		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 光标的图形宽度 (为 0 时, 赋光标位置字符的图形宽度)	8 位无符号	0, 255	图形像素
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ 本函数一定要在字符串或数值显示 (调用 F_DISPSTRA, F_DISPSTRB, F_DISPTXTA, F_DISPTXTB, F_DISPUIX16 或 F_DISPFIX16) 之后, 不改变活动区域之前调用, 否则, 光标的显示坐标可能与期望值不同. ◆ 光标的图形宽度和图形高度都不可能大于 24, 因此, 大于 24 的设置值都将被限制为 24. 			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETACTRECT ◆ F_DISPSTRA ◆ F_DISPSTRB ◆ F_DISPTXTA ◆ F_DISPTXTB ◆ F_DISPUIX16 ◆ F_DISPFIX16 			

功能名称:	F_SETBDPTR	功能命令代码:	24H
功能简介:	设置通用显示信息缓冲区数据指针		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ 通用显示信息缓冲区与自由存取区同样是随机存储器 (RAM) 内划出的一块区域, 而且都可以自由存取, 操作它们的功能命令也相对应, 唯一的区别是: 通用显示信息缓冲区可以作为某些点阵显示 (如: 位图显示, 字符串显示, 文本框显示等) 的数据源, 而自由存取区不行. ◆ 通用显示信息缓冲区, 自由存取区和可编程自由存取区的数据指针分别独立. ◆ 该功能命令只有在液晶的基本参数设置好之后执行才有效 (见 F_SETBASIC 功能). 			

- ◆ 输入参数是指通用显示信息缓冲区内的偏移地址, 外界无需关心缓冲区的首地址, 只需控制数据指针的值相对于在缓冲区首地址的偏移量即可。
- ◆ 输入参数必需小于通用显示信息缓冲区的大小, 否则不会改变数据指针。譬如: 我们在应用程序的开始执行 F_SETBASIC 功能命令, 已经将通用显示信息缓冲区的大小设置为 1K 字节, 则当输入参数 $[0^1] < 0400H$ 时是有效的, 否则便无效。
- ◆ 通用显示信息缓冲区的大小可以通过执行 F_GETBASIC 功能命令获得。

使用实例:

(略)

相关功能:

- ◆ F_GETBDPTR
- ◆ F_SETFDPTR
- ◆ F_SETEDPTR
- ◆ F_SETBASIC
- ◆ F_GETBASIC

功能名称:	F_GETBDPTR	功能命令代码:	25H
功能简介:	获取通用显示信息缓冲区的数据指针		
输入参数个数:	0	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0^1] 数据指针的值 (即缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			
(无)			
使用实例:			
(略)			
相关功能:			
◆ F_SETBDPTR			
◆ F_GETFDPTR			
◆ F_GETEDPTR			

功能名称:	F_WRBUFFA	功能命令代码:	28H
功能简介:	往通用显示信息缓冲区写一个数据, 数据指针循环增一		
输入参数个数:	1 (1)	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0^1] 数据指针的值 (即缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			

- ◆ 数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于通用显示信息缓冲区的大小则归零.
- ◆ 输出参数可用可不用. 它的作用是有时可以省去执行 F_GETBDPTR 功能命令的麻烦.

使用实例:

- ◆ 如果基于这样一个前提:通用显示信息缓冲区的大小为 0400H 字节, 当前数据指针的值为 03FFH, 输入参数为 AAH. 则执行该功能命令后, 数据缓冲区 03FFH 单元的值为 AAH, 数据指针的值变为 0000H. 请看以下一段 MCS51 的程序代码:

```
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
MOV      A,       #0AAH      ;[0]要写入的数据
MOVX     @DPTR,    A
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_WRBUFFA
MOVX     @DPTR,    A          ;往通用显示信息缓冲区写数据
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
MOVX     A,        @DPTR
MOV      R2,      A
MOVX     A,        @DPTR
MOV      R3,      A          ;R2`R3 = 0000H
```

相关功能:

- ◆ F_SETBDPTR
- ◆ F_GETBDPTR
- ◆ F_WRFREEA
- ◆ F_WREEPFA
- ◆ F_WRBUFFB
- ◆ F_WRBUFFC
- ◆ F_WRBUFFD

功能名称:	F_WRBUFFB	功能命令代码:	29H
功能简介:	往通用显示信息缓冲区写 N 个数据, 数据指针循环增加		
输入参数个数:	N+1 (N+1)	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
[0]	要写入的第 1 个数据	-	-
[1]	要写入的第 2 个数据	-	-
⋮	要写入的第 3 ~ N-1 个数据	-	-
[N-1]	要写入的第 N 个数据	-	-
[N]	要写入的数据个数 (N)	4 位无符号	0, 15
<3..0>			字节
输出参数说明:	参数编码方式	值域	单位

[0`1]	数据指针的值 (即缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:				
<ul style="list-style-type: none"> 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于通用显示信息缓冲区的大小则归零, 以此类推... 必须先按顺序送数据, 再送表示数据个数的参数. 参数[N]的值应与输入数据的个数相等, 否则执行结果将与预期的不符(请看使用实例). 输入数据最多为 15 个. 				
使用实例:				
<p>以下各例均基于这样一个前提:通用显示信息缓冲区的大小为 0400H 字节, 当前数据指针的值为 03FFH.</p> <ul style="list-style-type: none"> 如果输入参数为 5 个:[0]11H, [1]22H, [2]33H, [3]44H, [4]04H, 则通用信息缓冲区某些存储单元的值为:[03FFH]=11H, [0000H]=22H, [0001H]=33H, [0002H]=44H, 数据指针的值为 0003H. 如果输入参数为 5 个:[0]11H, [1]22H, [2]33H, [3]44H, [4]03H, 则通用信息缓冲区某些存储单元的值为:[03FFH]=22H, [0000H]=33H, [0001H]=44H, [0002H]不变, 数据指针的值为 0002H. 如果输入参数为 5 个:[0]11H, [1]22H, [2]33H, [3]44H, [4]05H, 则通用信息缓冲区某些存储单元的值为:[03FFH]不确定, [0000H]=11H, [0001H]=22H, [0002H]=33H, [0003H]=44H, 数据指针的值为 0004H. 				
相关功能:				
<ul style="list-style-type: none"> F_SETBDPTR F_GETBDPTR F_WRFREEB F_WREEPFB F_WRBUFFA F_WRBUFFC F_WRBUFFD 				

功能名称:	F_WRBUFFC	功能命令代码:	2AH
功能简介:	往通用显示信息缓冲区写 16 个数据, 数据指针循环增加		
输入参数个数:	16(16)	输出参数个数:	1(2)
输入参数说明:	参数编码方式	值域	单位
[0]	要写入的第 1 个数据	—	—
[1]	要写入的第 2 个数据	—	—
⋮	要写入的第 3 ~ 15 个数据	—	—
[15]	要写入的第 16 个数据	—	—
输出参数说明:	参数编码方式	值域	单位
[0`1]	数据指针的值 (即缓冲区内的偏移地址)	16 位无符号	0, 65535 字节
使用说明:			
<ul style="list-style-type: none"> 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于通用显示信息缓冲区的大小则归零, 以此类推... 输入参数必须是 16 个, 否则执行结果将与预期不符(参见 F_WRBUFFB 功能命令的使用实例部分). 			

使用实例：

- ◆ 如果没有外接 E²PROM, 则字符显示功能将无法使用, 但是可以用 F_WRBUFBB 和 F_WRBUFFC 功能命令将点阵信息送到通用信息缓冲区内, 然后再执行显示位图的命令来达到显示字符的目的. 比如显示一个 16×16 点阵的字符, 则可以用如下一段 MCS51 的代码实现:

```
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_GETBDPTR
MOVX     @DPTR,    A          ;获取通用显示信息缓冲区的数据指针
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
MOV      A,       @DPTR
MOV      R2,      A
MOV      A,       @DPTR
MOV      R3,      A          ;R2`R3=当前数据指针的值
MOV      A,       #02H
MOVX     @DPTR,    A          ;字符的文本宽度为 2
MOV      A,       #10H
MOVX     @DPTR,    A          ;字符的图形高度为 16
MOV      A,       #2
MOVX     @DPTR,    A          ;数据个数为 2
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_WRBUFBB
MOVX     @DPTR,    A          ;往通用显示信息缓冲区写 2 个数据
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
MOV      DPTR,    #LCD_DA    ;液晶的数据通道地址
;...                          ;送 16 个字节
MOV      DPTR,    #LCD_CA    ;液晶的命令通道地址
MOV      A,       #F_WRBUFFC
MOVX     @DPTR,    A          ;往通用显示信息缓冲区写 16 个数据
LCALL    WAIT_LCD_IDLE      ;等待液晶空闲
;...                          ;以同样的方法再写 16 个数据
;...                          ;执行显示位图的命令
```

以上只是表达了一种程序设计的思想, 实际应用中可以将其包装成显示字符的函数.

如果有外接 E²PROM, 仍然可以利用本例提供的设计思想实现字符的显示, 这种字符可能是没有预先存储在 E²PROM 中的.

相关功能：

- ◆ F_SETBDPTR
- ◆ F_GETBDPTR
- ◆ F_WRFREEC
- ◆ F_WREEPFC
- ◆ F_WRBUFFA
- ◆ F_WRBUFFB
- ◆ F_WRBUFFD

功能名称:	F_WRBUFFD	功能命令代码:	2BH
功能简介:	往通用显示信息缓冲区写一个数据,但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
[1`2] 写入地址(指缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ 输入的数据指针的值必须小于通用显示信息缓冲区的大小, 否则无效. ◆ 通用显示信息缓冲区的大小可以通过执行 F_GETBASIC 命令获得. ◆ 写入地址与当前数据指针没有关系. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 如原数据指针为 3, 写入地址为 8, 则数据写入缓冲区内偏移地址为 8 的单元, 数据指针仍然为 3. 此时若执行 F_WRBUFFA 命令将把新的数据送入缓冲区内偏移地址为 3 的单元. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_GETBASIC ◆ F_WRFREED ◆ F_WREPPFD ◆ F_WRBUFFA ◆ F_WRBUFFB ◆ F_WRBUFFC 			

功能名称:	F_RDBUFFB	功能命令代码:	2CH
功能简介:	从通用显示信息缓冲区读 N 个数据, 数据指针循环增加		
输入参数个数:	1(1)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1	4 位无符号	0, 15	字节
<3..0> (如 4: 表示读 5 个数据, N=5)			
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ N-1 个数据	-	-	-
[N-1] 读出的第 N 个数据	-	-	-
使用说明:			
(无)			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETBDPTR ◆ F_RDFREEB 			

◆ F_RDEEPPFB
◆ F_RDBUFFD

功能名称:	F_RDBUFFD	功能命令代码:	2DH
功能简介:	从通用显示信息缓冲区读 N 个数据, 但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1 <3..0> (如 4:表示读 5 个数据, N=5)	4 位无符号	0, 15	字节
[1`2] 起始读取地址 (缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ N-1 个数据	-	-	-
[N-1] 读出的第 N 个数据	-	-	-
使用说明:	<ul style="list-style-type: none">◆ 输入的数据指针的值必须小于通用显示信息缓冲区的大小, 否则读出的数据无效.◆ 通用显示信息缓冲区的大小可以通过执行 F_GETBASIC 命令获得.◆ 当输入的数据指针的值有效时, 读出当前数据指针指向的存储单元的内容, 然后数据指针增一, 如果数据指针的值等于通用显示信息缓冲区的大小则归零, 以此类推... 但是, 该功能命令执行完毕时, 数据指针恢复到执行前的值.◆ 起始读取地址的值与当前数据指针没有关系.		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none">◆ F_GETBASIC◆ F_RDFREED◆ F_RDEEPPD◆ F_RDBUFFB		

功能名称:	F_SETFDPTR	功能命令代码:	34H
功能简介:	设置自由存取区数据指针		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	<ul style="list-style-type: none">◆ (参见 F_SETBDPTR 功能命令的使用说明部分)		

◆ 自由存取区是在液晶显示模块的随机存储器内开辟的一块供外界自由存取数据的区域. 在访问速度没有要求的场合下, 该区域可以充当应用系统的随机存储器.
使用实例: (略)
相关功能: ◆ F_GETDPTR ◆ F_SETDPTR ◆ F_SETDPTR ◆ F_SETBASIC ◆ F_GETBASIC

功能名称:	F_GETDPTR	功能命令代码:	35H
功能简介:	获取自由存取区的数据指针		
输入参数个数:	0	输出参数个数:	1(2)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明: (无)			
使用实例: (略)			
相关功能: ◆ F_SETDPTR ◆ F_GETDPTR ◆ F_SETDPTR			

功能名称:	F_WRFREEA	功能命令代码:	38H
功能简介:	往自由存取区写一个数据, 数据指针循环增一		
输入参数个数:	1(1)	输出参数个数:	1(2)
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明: ◆ 数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于自由存取区的大小则归零. ◆ 输出参数可用可不用. 它的作用是有时可以省去执行 F_GETDPTR 功能命令的麻烦.			
使用实例:			

- ◆ 如果基于这样一个前提:自由存取区的大小为 0400H 字节, 当前数据指针的值为 03FFH, 输入参数为 AAH. 则执行该功能命令后, 自由存取区 03FFH 单元的值为 AAH, 数据指针的值变为 0000H. 请看以下一段 MCS51 的程序代码:

```
LCALL WAIT_LCD_IDLE ;等待液晶空闲
MOV DPTR, #LCD_DA ;液晶的数据通道地址
MOV A, #0AAH ;[0]要写入的数据
MOVX @DPTR, A
MOV DPTR, #LCD_CA ;液晶的命令通道地址
MOV A, #F_WRFREEA
MOVX @DPTR, A ;往自由存取区写数据
LCALL WAIT_LCD_IDLE ;等待液晶空闲
MOV DPTR, #LCD_DA ;液晶的数据通道地址
MOVX A, @DPTR
MOV R2, A
MOVX A, @DPTR
MOV R3, A ;R2`R3 = 0000H
```

相关功能:

- ◆ F_SETDPTR
- ◆ F_GETDPTR
- ◆ F_WRBUFA
- ◆ F_WREEPFA
- ◆ F_WRFREEB
- ◆ F_WRFREEC
- ◆ F_WRFREED

功能名称:	F_WRFREEB	功能命令代码:	39H
功能简介:	往自由存取区写 N 个数据, 数据指针循环增加		
输入参数个数:	N+1 (N+1)	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
[0]	要写入的第 1 个数据	—	—
[1]	要写入的第 2 个数据	—	—
⋮	要写入的第 3 ~ N-1 个数据	—	—
[N-1]	要写入的第 N 个数据	—	—
[N]	要写入的数据个数 (N)	4 位无符号	0, 15
<3..0>			字节
输出参数说明:	参数编码方式	值域	单位
[0`1]	数据指针的值 (即自由存取区内的偏移地址)	16 位无符号	0, 65535
			字节
使用说明:	<ul style="list-style-type: none"> ◆ 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于自由存取区的大小则归零, 以此类推... 		

- ◆ 必须先按顺序送数据, 再送表示数据个数的参数。
- ◆ 参数[N]的值应与输入数据的个数相等, 否则执行结果将与预期的不符(请看使用实例)。
- ◆ 输入数据最多为 15 个。

使用实例:

以下各例均基于这样一个前提: 自由存取区的大小为 0400H 字节, 当前数据指针的值为 03FFH。

- ◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]04H, 则自由存取区某些存储单元的值为: [03FFH]=11H, [0000H]=22H, [0001H]=33H, [0002H]=44H, 数据指针的值为 0003H。
- ◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]03H, 则自由存取区某些存储单元的值为: [03FFH]=22H, [0000H]=33H, [0001H]=44H, [0002H]不变, 数据指针的值为 0002H。
- ◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]05H, 则自由存取区某些存储单元的值为: [03FFH]不确定, [0000H]=11H, [0001H]=22H, [0002H]=33H, [0003H]=44H, 数据指针的值为 0004H。

相关功能:

- ◆ F_SETBDPTR
- ◆ F_GETBDPTR
- ◆ F_WRBUFBB
- ◆ F_WREEPFB
- ◆ F_WRFREEA
- ◆ F_WRFREEC
- ◆ F_WRFREED

功能名称:	F_WRFREEC	功能命令代码:	3AH
功能简介:	往自由存取区写 16 个数据, 数据指针循环增加		
输入参数个数:	16(16)	输出参数个数:	1(2)
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的第 1 个数据	-	-	-
[1] 要写入的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ 15 个数据	-	-	-
[15] 要写入的第 16 个数据	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			
◆ 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于自由存取区的大小则归零, 以此类推...			
◆ 输入参数必须是 16 个, 否则执行结果将与预期不符(参见 F_WRFREEB 功能命令的使用实例部分).			
使用实例:			
(略)			
相关功能:			
◆ F_SETDPTR			
◆ F_GETDPTR			
◆ F_WRBUFC			

- ◆ F_WREEPFC
- ◆ F_WRFREEA
- ◆ F_WRFREEB
- ◆ F_WRFREED

功能名称:	F_WRFREED	功能命令代码:	3BH
功能简介:	往自由存取区写一个数据,但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
[1`2] 写入地址 (指自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ 输入的数据指针的值必须小于自由存取区的大小, 否则无效. ◆ 自由存取区的大小可以通过执行 F_GETBASIC 命令获得. ◆ 写入地址与数据指针无关. ◆ 自由存取区的数据指针与通用信心缓冲区的数据指针相互独立. 			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> ◆ F_GETBASIC ◆ F_WRBUFFD ◆ F_WREEPFD ◆ F_WRFREEA ◆ F_WRFREEB ◆ F_WRFREED 			

功能名称:	F_RDFREEB	功能命令代码:	3CH
功能简介:	从自由存取区读 N 个数据, 数据指针循环增加		
输入参数个数:	1(1)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1 <3..0> (如 4: 表示读 5 个数据, N=5)	4 位无符号	0, 15	字节
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ N-1 个数据	-	-	-
[N-1] 读出的第 N 个数据	-	-	-
使用说明:			

(无)
使用实例:
(略)
相关功能:
<ul style="list-style-type: none"> ◆ F_SETFDPTR ◆ F_RDBUFFB ◆ F_RDEEPPFB ◆ F_RDFREED

功能名称:	F_RDFREED	功能命令代码:	3DH
功能简介:	从自由存取区读 N 个数据, 但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1	4 位无符号	0, 15	字节
<3..0> (如 4: 表示读 5 个数据, N=5)			
[1`2] 起始读取地址	16 位无符号	0, 65535	字节
(指自由存取区内的偏移地址)			
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ N-1 个数据	-	-	-
[N-1] 读出的第 N 个数据	-	-	-
使用说明:	<ul style="list-style-type: none"> ◆ 输入的数据指针的值必须小于自由存取区的大小, 否则读出的数据无效. ◆ 当输入的数据指针的值有效时, 读出当前数据指针指向的存储单元的内容, 然后数据指针增一, 如果数据指针的值等于自由存取区的大小则归零, 以此类推... 但是, 该功能命令执行完毕时, 数据指针恢复到执行前的值. ◆ 起始读取地址与数据指针无关. 		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_GETBASIC ◆ F_RDBUFFD ◆ F_RDEEPPFD ◆ F_RDFREEB 		

功能名称:	F_EEPWREN	功能命令代码:	42H
功能简介:	设置可编程自由存取区的写保护开关		
输入参数个数:	1(1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位

[0] 写保护开关: 0:允许写, 非 0:写保护			
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ 液晶上电复位后, 可编程自由存取区处于写保护状态, 只可读不可写.			
使用实例:			
(略)			
相关功能:			
◆ F_GETLSW			

功能名称:	F_SETEDPTR	功能命令代码:	44H
功能简介:	设置可编程自由存取区数据指针		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ (参见 F_SETBDPTR 功能命令的使用说明部分)			
◆ 可编程自由存取区是在液晶显示模块的 E ² PROM 内开辟的一块供外界自由存取数据的区域. 在访问速度没有要求的场合下, 该区域可以充当应用系统的可编程只读存储器, 即使掉电, 数据也不会丢失. 很多仪器仪表都需要保存系统状态或参数的设置信息, 因此可编程自由存取区是非常实用的.			
◆ 无论可编程自由存取区是否写保护, 只要 E ² PROM 存在且被预置过, 该功能总是有效的.			
使用实例:			
(略)			
相关功能:			
◆ F_GETEDPTR			
◆ F_SETBDPTR			
◆ F_SETFDPTR			

功能名称:	F_GETEDPTR	功能命令代码:	45H
功能简介:	获取可编程自由存取区的数据指针		
输入参数个数:	0	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值 (即可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			

(无)
使用实例:
(略)
相关功能:
<ul style="list-style-type: none"> ◆ F_SETDPTR ◆ F_GETBDPTR ◆ F_GETFDPTR

功能名称:	F_WREPPFA	功能命令代码:	48H
功能简介:	往可编程自由存取区写一个数据, 数据指针循环增一		
输入参数个数:	1 (1)	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值(即可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			
<ul style="list-style-type: none"> ◆ 数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于可编程自由存取区的大小则归零. ◆ 输出参数可用可不用. 它的作用是有时可以省去执行 F_GETEDPTR 功能命令的麻烦. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 如果基于这样一个前提: 可编程自由存取区的大小为 0400H 字节, 当前数据指针的值为 03FFH, 输入参数为 AAH. 则执行该功能命令后, 可编程自由存取区 03FFH 单元的值为 AAH, 数据指针的值变为 0000H. 请看以下一段 MCS51 的程序代码: <pre> LCALL WAIT_LCD_IDLE ;等待液晶空闲 MOV DPTR, #LCD_DA ;液晶的数据通道地址 MOV A, #0AAH ;[0]要写入的数据 MOVX @DPTR, A MOV DPTR, #LCD_CA ;液晶的命令通道地址 MOV A, #F_WREPPFA MOVX @DPTR, A ;往可编程自由存取区写数据 LCALL WAIT_LCD_IDLE ;等待液晶空闲 MOV DPTR, #LCD_DA ;液晶的数据通道地址 MOVX A, @DPTR MOV R2, A MOVX A, @DPTR MOV R3, A ;R2`R3 = 0000H </pre>			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETDPTR ◆ F_GETEDPTR ◆ F_WRBUFFA ◆ F_WRFREEA 			

◆ F_WREEPFB
◆ F_WREEPFC
◆ F_WREEPFD

功能名称:	F_WREEPFB	功能命令代码:	49H
功能简介:	往可编程自由存取区写 N 个数据, 数据指针循环增加		
输入参数个数:	N+1 (N+1)	输出参数个数:	1 (2)
输入参数说明:	参数编码方式	值域	单位
[0]	要写入的第 1 个数据	—	—
[1]	要写入的第 2 个数据	—	—
⋮	要写入的第 3 ~ N-1 个数据	—	—
[N-1]	要写入的第 N 个数据	—	—
[N]	要写入的数据个数 (N)	4 位无符号	0, 15
<3..0>			字节
输出参数说明:	参数编码方式	值域	单位
[0`1]	数据指针的值(即可编程自由存取区内的偏移地址)	16 位无符号	0, 65535
			字节
使用说明:			
◆ 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于可编程自由存取区的大小则归零, 以此类推...			
◆ 必须先按顺序送数据, 再送表示数据个数的参数.			
◆ 参数[N]的值应与输入数据的个数相等, 否则执行结果将与预期的不符(请看使用实例).			
◆ 输入数据最多为 15 个.			
使用实例:			
以下各例均基于这样一个前提: 可编程自由存取区的大小为 0400H 字节, 当前数据指针的值为 03FFH.			
◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]04H, 则可编程自由存取区某些存储单元的值为: [03FFH]=11H, [0000H]=22H, [0001H]=33H, [0002H]=44H, 数据指针的值为 0003H.			
◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]03H, 则可编程自由存取区某些存储单元的值为: [03FFH]=22H, [0000H]=33H, [0001H]=44H, [0002H]不变, 数据指针的值为 0002H.			
◆ 如果输入参数为 5 个: [0]11H, [1]22H, [2]33H, [3]44H, [4]05H, 则可编程自由存取区某些存储单元的值为: [03FFH]不确定, [0000H]=11H, [0001H]=22H, [0002H]=33H, [0003H]=44H, 数据指针的值为 0004H.			
相关功能:			
◆ F_SETDPTR			
◆ F_GETDPTR			
◆ F_WRBUFEB			
◆ F_WRFREEB			
◆ F_WREEPFA			
◆ F_WREEPFC			
◆ F_WREEPFD			

功能名称:	F_WREEPFC	功能命令代码:	4AH
功能简介:	往可编程自由存取区写 16 个数据, 数据指针循环增加		
输入参数个数:	16(16)	输出参数个数:	1(2)
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的第 1 个数据	-	-	-
[1] 要写入的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ 15 个数据	-	-	-
[15] 要写入的第 16 个数据	-	-	-
输出参数说明:	参数编码方式	值域	单位
[0`1] 数据指针的值(即可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
使用说明:			
<ul style="list-style-type: none"> 第 1 个数据写入当前数据指针指向的存储单元, 并且数据指针的值增一, 如果数据指针的值等于可编程自由存取区的大小则归零, 以此类推... 输入参数必须是 16 个, 否则执行结果将与预期不符(参见 F_WREEPFB 功能命令的使用实例部分)。 			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> F_SETEDPTR F_GETEDPTR F_WRBUFFC F_WRFREEC F_WREEPFA F_WREEPFB F_WREEPFD 			

功能名称:	F_WREEPFD	功能命令代码:	4BH
功能简介:	往可编程自由存取区写一个数据, 但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 要写入的数据	-	-	-
[1`2] 写入地址(指可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:			
<ul style="list-style-type: none"> 输入的数据指针的值必须小于可编程自由存取区的大小, 否则无效。 可编程自由存取区的大小可以通过执行 F_GETBASIC 命令获得。 写入地址与数据指针无关。 可编程自由存取区与通用信息缓冲区和自由存取区的数据指针相互独立。 			
使用实例:			

(略)
相关功能:
<ul style="list-style-type: none"> ◆ F_GETBASIC ◆ F_WRBUFFD ◆ F_WREPPFD ◆ F_WRFREEA ◆ F_WRFREEB ◆ F_WRFREEC

功能名称:	F_RDEEPPB	功能命令代码:	4CH
功能简介:	从可编程自由存取区读 N 个数据, 数据指针循环增加		
输入参数个数:	1(1)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1 <3..0> (如 4:表示读 5 个数据, N=5)	4 位无符号	0, 15	字节
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-
⋮ 要写入的第 3 ~ N-1 个数据	-	-	-
[N-1] 读出的第 N 个数据	-	-	-
使用说明:	(无)		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_SETDPTR ◆ F_RDBUFFB ◆ F_RDFREEB ◆ F_RDEEPPD 		

功能名称:	F_RDEEPPD	功能命令代码:	4DH
功能简介:	从可编程自由存取区读 N 个数据, 但不改变数据指针		
输入参数个数:	2(3)	输出参数个数:	N(N)
输入参数说明:	参数编码方式	值域	单位
[0] 数据个数(N)-1 <3..0> (如 4:表示读 5 个数据, N=5)	4 位无符号	0, 15	字节
[1`2] 起始读取地址(指可编程自由存取区内的偏移地址)	16 位无符号	0, 65535	字节
输出参数说明:	参数编码方式	值域	单位
[0] 读出的第 1 个数据	-	-	-
[1] 读出的第 2 个数据	-	-	-

⋮	要写入的第 3 ~ N-1 个数据	—	—	—
[N-1]	读出的第 N 个数据	—	—	—
使用说明:				
<ul style="list-style-type: none"> ◆ 输入的数据指针的值必须小于可编程自由存取区的大小, 否则读出的数据无效. ◆ 可编程自由存取区的大小可以通过执行 F_GETBASIC 命令获得. ◆ 当输入的数据指针的值有效时, 读出当前数据指针指向的存储单元的内容, 然后数据指针增一, 如果数据指针的值等于可编程自由存取区的大小则归零, 以此类推... 但是, 该功能命令执行完毕时, 数据指针恢复到执行前的值. ◆ 起始读取地址与数据指针无关. 				
使用实例:				
(略)				
相关功能:				
<ul style="list-style-type: none"> ◆ F_GETBASIC ◆ F_RDBUFFD ◆ F_RDFREED ◆ F_RDEEPPFB 				

功能名称:	F_COPYPAGE	功能命令代码:	50H
功能简介:	复制整个页面		
输入参数个数:	2 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 目标页面的序号	8 位无符号	0, 7	—
[1] 源页面的序号	8 位无符号	0, 7	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:			
<ul style="list-style-type: none"> ◆ 两个输入参数的值都应小于页面数目, 否则操作无效. ◆ 复制整个页面是比较慢的操作, 除非必要, 建议使用 F_SAVEACTRECT 功能实现小范围的复制. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 这个功能的一般应用是避免显示的过渡过程, 比如: 我们把第 0 页设置成显示页面, 第 1 页设置成操作页面, 在操作页面上先进行复杂的显示, 再将第 1 页的显示信息复制到第 0 页上. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SAVEACTRECT ◆ F_RESUACTRECT ◆ F_SETPAGEINFO ◆ F_GETPAGEINFO ◆ F_GETBASIC ◆ F_SETBASIC 			

功能名称:	F_SAVEACTRECT	功能命令代码:	51H
功能简介:	保存活动区域的内容到其它页面		


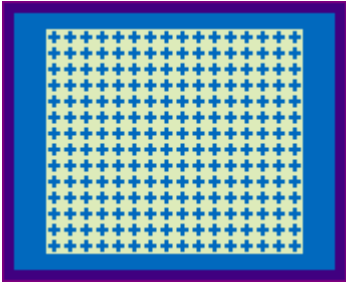
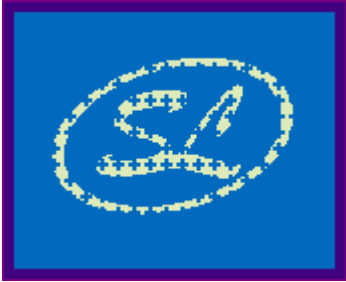
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 目标页面的序号	8 位无符号	0, 7	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none">◆ 输入参数的值应小于页面数目, 否则操作无效.◆ 该功能是将操作页面内可视活动区域的显示内容复制到其它页面的相同区域内. 与 F_COPYPAGE 不同的是: 该功能的源页面始终是操作页面. 因为活动区域只在操作页面中存在.			
使用实例:			
<ul style="list-style-type: none">◆ 下例基于这个前提: 第 0 页为显示页面和操作页面 <div><pre>graph TD; Start(()) --> Step1[将活动区域的内容复制到第1页]; Step1 --> Step2[将第1页设置为操作页面]; Step2 --> Step3[过程复杂的显示]; Step3 --> Step4[将活动区域的内容复制到第0页]; Step4 --> Step5[将第0页设置为操作页面]; Step5 --> End(())</pre></div> <p>这个例子可以实现动画, 文字移动等操作的一个循环, 即循环执行上面的流程可以制作出动画, 字幕等效果.</p>			
相关功能:			
<ul style="list-style-type: none">◆ F_RESUACTRECT◆ F_COPYPAGE◆ F_SETPAGEINFO◆ F_GETPAGEINFO◆ F_SETACTRECT◆ F_GETBASIC◆ F_SETBASIC			

功能名称:	F_RESUACTRECT	功能命令代码:	52H
功能简介:	从其它页面中恢复活动区域的内容		
输入参数个数:	1 (1)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 源页面的序号	8 位无符号	0, 7	-
输出参数说明:	参数编码方式	值域	单位

使用说明:
◆ 输入参数的值应小于页面数目, 否则操作无效.
使用实例:
◆ 该功能命令与 F_SAVEACTRECT 功能配合能够实现弹出式菜单的背景恢复, 但是一定要注意活动区域应与保存时相同.
相关功能:
◆ F_SAVEACTRECT
◆ F_COPYPAGE
◆ F_SETPAGEINFO
◆ F_GETPAGEINFO
◆ F_GETBASIC
◆ F_SETBASIC

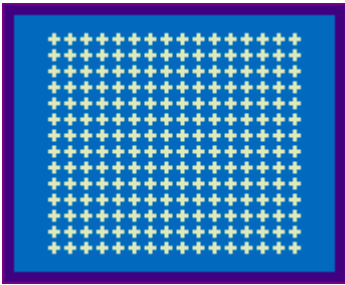
功能名称:	F_CLRSCREEN	功能命令代码:	56H
功能简介:	清除整个操作页面的显示内容		
输入参数个数:	0	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
-	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	◆ 该功能属于点阵显示, 但是它只有普通显示方式, 没有逻辑显示方式. ◆ 该功能受到反显方式的影响, 它的效果等价于将反显方式字节添满整个操作页面.		
使用实例:	(略)		
相关功能:	◆ F_SETINVERT ◆ F_FILLRECT		

功能名称:	F_FILLRECT	功能命令代码:	58H
功能简介:	填充矩形区域		
输入参数个数:	6(8)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1]	矩形区域的左上角文本横坐标	16 位补码	-1024, 1023 文本像素
[2`3]	矩形区域的左上角图形纵坐标	16 位补码	-1024, 1023 图形像素
[4]	矩形区域的文本宽度	8 位无符号	0, 255 文本像素
[5]	矩形区域的图形高度	8 位无符号	0, 255 图形像素
[6]	填充模板编号	8 位无符号	0, 255 -

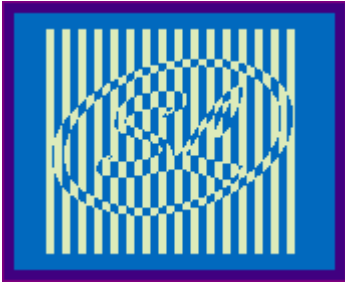
[7] 显示方式				
<2..0> 0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或		3 位无符号	0, 5	-
输出参数说明:		参数编码方式	值域	单位
-		-	-	-
使用说明:				
<ul style="list-style-type: none">◆ 输入坐标会被自动限制在值域内.◆ 输入坐标值是以活动区域左上角为参照的.◆ 实际填充的区域受可视活动区域的限制.◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响.◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响.				
使用实例:				
<ul style="list-style-type: none">◆ (普通显示方式请参看 F_SETFILTER 的使用实例) <p>逻辑显示方式例子的前提是: 页面文本宽度为 20, 图形高度为 128, 活动区域为整个页面, 页面上已经显示了一幅位图.</p>				
				
<ul style="list-style-type: none">◆ 输入参数: [0]00H, [1]02H, [2]00H, [3]08H, [4]10H, [5]70H, [6]99H, [7]01H. 执行结果为:				
				
<ul style="list-style-type: none">◆ 输入参数: [0]00H, [1]02H, [2]00H, [3]08H, [4]10H, [5]70H, [6]99H, [7]02H. 执行结果为:				
				
<ul style="list-style-type: none">◆ 输入参数: [0]00H, [1]02H, [2]00H, [3]08H, [4]10H, [5]70H, [6]E2H, [7]03H. 执行结果为:				



◆ 输入参数:[0]00H, [1]02H, [2]00H, [3]08H, [4]10H, [5]70H, [6]99H, [7]04H. 执行结果为:



◆ 输入参数:[0]00H, [1]02H, [2]00H, [3]08H, [4]10H, [5]70H, [6]34H, [7]05H. 执行结果为:



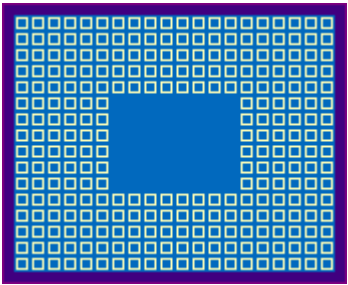
◆ 选择一系列填充模板, 并使用逻辑与方式进行填充, 可以实现背景淡出的效果.

相关功能:	
◆ F_FILLACTRECT	
◆ F_FILLRECTOUT	
◆ F_CLRSCREEN	
◆ F_SETACTRECT	
◆ F_SETFILTER	
◆ F_SETINVERT	
◆ F_SETUNDERLINE	

功能名称:	F_FILLACTRECT	功能命令代码:	59H
功能简介:	填充整个活动区域(其实是填充整个可视活动区域)		
输入参数个数:	2(2)	输出参数个数:	0
输入参数说明:		参数编码方式	值域
[0] 填充模板编号		-	-

[1]	显示方式			
<2..0>	0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或	3 位无符号	0, 5	-
输出参数说明:		参数编码方式	值域	单位
-		-	-	-
使用说明:				
(无)				
使用实例:				
◆ (参看 F_FILLRECT 功能命令的使用实例)				
相关功能:				
◆ F_FILLRECT				
◆ F_CLRSCREEN				
◆ F_SETACTRECT				
◆ F_SETFILTER				
◆ F_SETINVERT				
◆ F_SETUNDERLINE				

功能名称:	F_FILLRECTOUT	功能命令代码:	5AH	
功能简介:	填充矩形以外的区域			
输入参数个数:	6(8)	输出参数个数:	0	
输入参数说明:	参数编码方式	值域	单位	
[0`1]	矩形区域的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	矩形区域的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4]	矩形区域的文本宽度	8 位无符号	0, 255	文本像素
[5]	矩形区域的图形高度	8 位无符号	0, 255	图形像素
[6]	填充模板编号	8 位无符号	0, 255	-
[7]	显示方式			
<2..0>	0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或	3 位无符号	0, 5	-
输出参数说明:	参数编码方式	值域	单位	
-	-	-	-	-
使用说明:				
◆ (参见 F_FILLRECT 的使用说明)				
使用实例:				
◆ 前提:页面文本宽度和图形高度分别为 20 和 128, 当前活动区域为整个页面, 页面上没有显示内容. 若输入参数为: [0]00H, [1]06H, [2]00H, [3]28H, [4]08H, [5]30H, [6]84H, [7]01H. 执行结果为:				

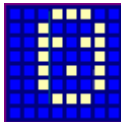


相关功能:

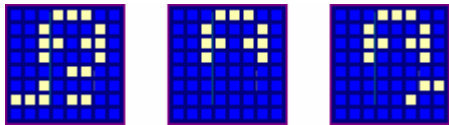
- ◆ F_FILLACTRECT
- ◆ F_FILLRECT
- ◆ F_CLRSCREEN
- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE

功能名称:		F_DISPBYTE		功能命令代码:		5DH			
功能简介:		显示文本像素							
输入参数个数:		4 (6)		输出参数个数:		0			
输入参数说明:				参数编码方式		值域		单位	
[0`1]		文本横坐标		16 位补码		-1024, 1023		文本像素	
[2`3]		图形纵坐标		16 位补码		-1024, 1023		图形像素	
[4]		文本像素字节		-		-		-	
[5]		显示方式							
<2..0>		0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或		3 位无符号		0, 5		-	
输出参数说明:				参数编码方式		值域		单位	
-		-		-		-		-	
使用说明:									
<ul style="list-style-type: none">◆ 输入坐标会被自动限制在值域内.◆ 必须输入正确的显示方式, 否则无法显示.◆ 输入坐标值是以活动区域左上角为参照的.◆ 显示受活动区域的限制.◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响.◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响.◆ 虽然可以用该功能命令实现位图或字符的显示, 但是建议不要这样做, 因为这样做的显示速度远比直接的位图显示要慢. 显示字符的方法请参看 F_WRBUFC 功能命令.									
使用实例:									
<ul style="list-style-type: none">◆ (逻辑显示方式请参看 F_FILLRECT 功能命令的使用实例)									
相关功能:									
<ul style="list-style-type: none">◆ F_FILLRECT◆ F_SETCURSOR									

- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE

功能名称:		F_DISPMPA		功能命令代码:		5EH	
功能简介:		显示通用显示信息缓冲区内的位图					
输入参数个数:		5(7)		输出参数个数:		0	
输入参数说明:				参数编码方式	值域	单位	
[0`1]		位图的左上角文本横坐标		16 位补码	-1024, 1023	文本像素	
[2`3]		位图的左上角图形纵坐标		16 位补码	-1024, 1023	图形像素	
[4`5]		位图点阵信息的首地址(在通用显示信息缓冲区内的偏移地址)		16 位无符号	0, 65535	字节	
[6]		显示方式					
[2..0]		0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或非		3 位无符号	0, 5	-	
[6]<4>		0:清除当前活动区域内位图以外的所有其它已显示的内容		-	-	-	
输出参数说明:				参数编码方式	值域	单位	
-		-		-	-	-	
使用说明:							
<ul style="list-style-type: none">◆ 显示通用显示信息缓冲区内的位图, 必须执行 F_DISPMPA 功能命令, 而不是 F_DISPMPB.◆ 输入坐标会被自动限制在值域内.◆ 必须输入正确的显示方式, 否则无法显示.◆ 输入坐标值是以活动区域左上角为参照的.◆ 显示受活动区域的限制.◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响.◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响.◆ 必须保证位图点阵信息的首地址小于通用显示信息缓冲区的大小, 并且它的值是正确的.◆ 还应该注意的是:通用显示信息缓冲区的大小足够存放所需的位图点阵信息, 并且点阵信息不出现循环存放的现象(即存放过程中不象 F_WRBUIFB 功能中提到的数据指针复位的情况). 否则, 显示结果将与预期结果产生出入.							
使用实例:							
<ul style="list-style-type: none">◆ (逻辑显示方式请参看 F_FILLRECT 功能命令的使用实例)◆ 如果通用显示信息缓冲区的大小为 0100H, 缓冲区内以下存储单元的值分别为: [00F6H]=01H, [00F7H]=08H, [00F8H]=1CH, [00F9H]=22H, [00FAH]=36H, [00FBH]=22H, [00FCH]=2AH, [00FDH]=22H, [00FEH]=1CH, [00FFH]=00H. 从[00F6H]单元开始的 10 个字节即为下面这个位图的点阵信息(其中头两个字节分别表示位图的文本宽度和图形高度):							
							
如果操作页面为显示页面, 输入参数[4`5]=00F6H, [6]=01H, 坐标参数合理使之完整显示的话, 我们便可在液晶屏上看到上面这个位图.							

但是,如果在送位图点阵信息的时候没有注意到缓冲区当前数据指针的位置时,很可能会出现这样一个存储结果:[00FAH]=01H, [00FBH]=08H, [00FCH]=1CH, [00FDH]=22H, [00FEH]=36H, [00FFH]=22H, [0000H]=2AH, [0001H]=22H, [0002H]=1CH, [0003H]=00H. 因为数据指针是循环增加的. 此时如果输入参数[4`5]改为 00FAH, 则显示结果可能为:



即后 4 行的数据是不定的.

◆ 参数[6]<4>的使用方法, 请参看 F_DISPBPMB 功能命令中关于参数[6]<4>的实例.

相关功能:

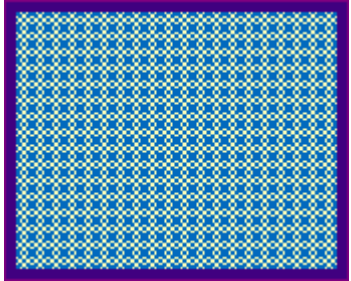
- ◆ F_DISPBPMB
- ◆ F_SETBDPTR
- ◆ F_GETBDPTR
- ◆ F_WRBUFBB
- ◆ F_WRBUFFC
- ◆ F_SETCURSOR
- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE

功能名称:	F_DISPBPMB	功能命令代码:	5FH	
功能简介:	显示预置的位图			
输入参数个数:	5(7)	输出参数个数:	0	
输入参数说明:		参数编码方式	值域	单位
[0`1]	位图的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	位图的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	位图点阵信息的首地址(在 E ² PROM 预置数据存储区内的偏移地址)	16 位无符号	0, 65535	字节
[6]	显示方式			
<2..0>	0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或非	3 位无符号	0, 5	-
[6]<4>	非 0:清除当前活动区域内位图以外的所有其它已显示的内容	-	-	-
输出参数说明:		参数编码方式	值域	单位
-	-	-	-	-
使用说明:				
◆ 显示预置在 E ² PROM 中的位图, 必须执行 F_DISPBPMB 功能命令, 而不是 F_DISPBPMA.				
◆ 输入坐标会被自动限制在值域内.				
◆ 必须输入正确的显示方式, 否则无法显示.				
◆ 输入坐标值是以活动区域左上角为参照的.				
◆ 显示受活动区域的限制.				
◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响.				

- ◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响.
- ◆ 必须保证位图点阵信息的首地址是正确的.

使用实例:

- ◆ (逻辑显示方式请参看 F_FILLRECT 功能命令的使用实例)
- ◆ 若页面文本宽度为 20, 图形高度为 128, 活动区域的左上角坐标为 (1, 8), 活动区域的文本宽度和图形高度分别为 18 和 112; 并且, 页面上已经显示了这样的图案:



参数[6]<4>为 0 时, 显示某个位图效果如下:



若其余参数不变, 仅将参数[6]<4>改为 1, 则显示效果如下:



- ◆ 利用参数[6]<4>为 1 时产生的效果, 可以很容易实现位图在某个活动区域内的移动.

相关功能:

- ◆ F_DISPBMPI
- ◆ F_SETCURSOR
- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE

功能名称:	F_DISPCHR	功能命令代码:	60H
功能简介:	显示字符		

输入参数个数:	4 (7)	输出参数个数:	0								
输入参数说明:	参数编码方式	值域	单位								
[0`1] 字符的左上角文本横坐标	16 位补码	-1024, 1023	文本像素								
[2`3] 字符的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素								
[4`5] 字符编码(请参考使用说明)	16 位无符号	0, 65535	-								
[6] 显示方式											
<2..0> 0:普通方式; 1:复制, 2:逻辑与, 3:逻辑或, 4:逻辑非, 5:逻辑异或	3 位无符号	0, 5	-								
输出参数说明:	参数编码方式	值域	单位								
-	-	-	-								
使用说明:											
<ul style="list-style-type: none"> ◆ 该功能命令只有在 E²PROM 被预置过才支持自定义字符的显示. ◆ 只要中英文全字库 ROM 存在, 该功能便可显示标准字符. ◆ 输入坐标会被自动限制在值域内. ◆ 必须输入正确的显示方式, 否则无法显示. ◆ 输入坐标值是以活动区域左上角为参照的. ◆ 显示受活动区域的限制. ◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响. ◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响. ◆ 字符编码规范: <ol style="list-style-type: none"> 1. 基本原则 <p>每一个字符编码用 16 位二进制数表示.</p> <p>0000h ~ 7FFFh 范围的编码为自定义字库使用;</p> <p>8000h ~ FFFFh 范围的编码为全字库使用.</p> 2. 自定义字符的编码规范 <p>自定义字符的编码规范表示如下:</p> <table border="1" data-bbox="392 1317 1007 1411"> <tr> <td>0</td> <td>LibIndex</td> <td>ChrIndex</td> <td>←符号</td> </tr> <tr> <td>1</td> <td>4</td> <td>11</td> <td>←位数</td> </tr> </table> <p>LibIndex: 自定义字库的编号; 值范围为 0h ~ Fh.</p> <p>ChrIndex: 字符在字库中的序号; 值范围为 000h ~ 7FFh. 特别地, 序号为 0 的字符为空白字符.</p> 3. 标准英文字符的编码规范 <p>标准英文字符的编码范围为 8000h ~ 80FFh, 分别对应 ASCII 编码 00h ~ FFh 的字符.</p> 4. 标准中文字符的编码规范 <p>标准中文字符的编码范围为 A1A0 ~ A1FF, A2A0 ~ A2FF, ..., F6A0 ~ F6FF, F7A0 ~ F7FF, 与国标一致.</p> 5. 控制字符 <p>控制字符统一只有一个, 即换行符, 编码为 FFFFh. 无论字库是否存在, 该字符总是存在的.</p> ◆ 编码无效的字符不会被显示出来. ◆ 标准字符的点阵类型为 16×16. 				0	LibIndex	ChrIndex	←符号	1	4	11	←位数
0	LibIndex	ChrIndex	←符号								
1	4	11	←位数								
使用实例:											
<ul style="list-style-type: none"> ◆ (逻辑显示方式请参看 F_FILLRECT 功能命令的使用实例) ◆ 如果中英文字库 ROM 存在, 我们想在 (2, 6) 坐标处显示标准字符“啊”, 则可以使用以下一段 MCS51 											

的代码实现：



```
LCALL  WAIT_LCD_IDLE      ;等待液晶空闲
MOV    DPTR,  #LCD_DA      ;液晶的数据通道地址
MOV    A,     #00H          ;[0`1] 文本横坐标
MOVX   @DPTR, A
MOV    A,     #02H
MOVX   @DPTR, A
MOV    A,     #00H          ;[2`3] 图形纵坐标
MOVX   @DPTR, A
MOV    A,     #06H
MOVX   @DPTR, A
MOV    A,     #0B0H         ;[4`5] 字符编码
MOVX   @DPTR, A
MOV    A,     #0A1H
MOVX   @DPTR, A
MOV    A,     #01H          ;[6] 复制显示方式
MOVX   @DPTR, A
MOV    DPTR,  #LCD_CA      ;液晶的命令通道地址
MOV    A,     #F_DISPCHR
MOVX   @DPTR, A            ;显示字符
LCALL  WAIT_LCD_IDLE      ;等待液晶空闲
```

◆ 上例中如果改成显示标准英文字符“A”，则字符编码参数[4`5]改成 8041H 即可。

相关功能：



- ◆ F_SETCURSOR
- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE

功能名称：	F_DISPSTRA	功能命令代码：	61H	
功能简介：	显示通用显示信息缓冲区内的字符串			
输入参数个数：	7(9)	输出参数个数：	4(6)	
输入参数说明：		参数编码方式	值域	单位
[0`1]	字符串的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	字符串的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	字符串信息的首地址(在通用显示信息缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
[6]	行高(相邻两行顶端之间的间距)	7 位无符号	0, 127	-
<6..0>	显示方式			
[7]	0:普通方式；1:复制，2:逻辑与，	3 位无符号	0, 5	-
<2..0>	3:逻辑或，4:逻辑非，5:逻辑异或			

[7]`<4>	1:清除当前活动区域内字符串以外的所有其它已显示的内容	-	-	-												
[8]	光标所在字符位的序号	8 位无符号	0, 255	-												
输出参数说明:		参数编码方式	值域	单位												
[0`1]	光标处字符的左上角文本横坐标	16 位补码	-1024, 1023	文本像素												
[2`3]	光标处字符的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素												
[4]	光标处字符的文本宽度	8 位无符号	0, 255	文本像素												
[5]	光标处字符的的图形高度	8 位无符号	0, 255	图形像素												
使用说明:																
<ul style="list-style-type: none">◆ 输入坐标会被自动限制在值域内.◆ 必须输入正确的显示方式, 否则无法显示.◆ 输入坐标值是以活动区域左上角为参照的.◆ 显示受活动区域的限制.◆ 普通显示方式受过滤器, 反显方式和下划线方式的影响.◆ 逻辑显示方式不受过滤器, 反显方式和下划线方式的影响.◆ 字符串的格式:																
<table><tr><td>StrL</td><td>Chr₀</td><td>Chr₁</td><td>.....</td><td>Chr_{StrL-1}</td><td>←符号</td></tr><tr><td>1</td><td>2</td><td>2</td><td>.....</td><td>2</td><td>←字节数</td></tr></table> <p>StrL:表示字符串的长度(字符数, 0≤StrL≤255)</p> <p>Chr_N:字符编码(N=0, 1,, StrL-1)</p> <ul style="list-style-type: none">◆ 字符串中可以包含自定义字符, 标准字符和控制字符, 但是在显示时它们是顶端对齐的.◆ 必须保证字符串信息的首地址是正确的, 否则显示结果将和预期结果不符(参见 F_DISPBMFA 中的使用实例).◆ 假设字符串长度为 m, 光标显示在字符串的第 n 个字符上(n < m), 则光标所在字符位的序号即为 n.◆ 利用参数[8]和功能命令 F_MODISTRUCUR 有助于实现文本编辑效果.					StrL	Chr ₀	Chr ₁	Chr _{StrL-1}	←符号	1	2	2	2	←字节数
StrL	Chr ₀	Chr ₁	Chr _{StrL-1}	←符号											
1	2	2	2	←字节数											
使用实例:																
以下举一些字符串显示效果的例子(假设活动区域和可视活动区域的文本宽度均为 24, 图形高度均为 80):																
<ul style="list-style-type: none">◆ 																
(本例中, 字符串的显示坐标为 (0, 0), 行高为 36, 其中 “,” 号后面是一个换行控制符)																
<ul style="list-style-type: none">◆ 																
(本例中, 字符串的显示坐标为 (0, 0), 行高为 36, 其中 “,” 号后面没有换行控制符)																
<ul style="list-style-type: none">◆ 参数[7]`<4>的使用方法, 请参看 F_DISPBMFA 功能命令中关于参数[7]`<4>的实例.																
相关功能:																
<ul style="list-style-type: none">◆ F_DISPSTRB◆ F_DISPTXTA◆ F_DISPTXTB																

◆ F_SETCURSOR
◆ F_SETACTRECT
◆ F_SETFILTER
◆ F_SETINVERT
◆ F_SETUNDERLINE
◆ F_MODISTRCUR

功能名称:	F_DISPSTRB	功能命令代码:	62H	
功能简介:	显示预置的字符串			
输入参数个数:	7(9)	输出参数个数:	4(6)	
输入参数说明:	参数编码方式	值域	单位	
[0`1]	字符串的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	字符串的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	字符串信息的首地址(在 E ² PROM 预置数据存储区内的偏移地址)	16 位无符号	0, 65535	字节
[6]	行高(相邻两行顶端之间的间距)	7 位无符号	0, 127	-
<6..0>	显示方式			
[7]	0:普通方式; 1:复制, 2:逻辑与,	3 位无符号	0, 5	-
<2..0>	3:逻辑或, 4:逻辑非, 5:逻辑异或非			
[7]<4>	0:清除当前活动区域内字符串以外的所有其它已显示的内容	-	-	-
[8]	光标所在字符位的序号	8 位无符号	0, 255	-
输出参数说明:	参数编码方式	值域	单位	
[0`1]	光标处字符的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	光标处字符的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4]	光标处字符的文本宽度	8 位无符号	0, 255	文本像素
[5]	光标处字符的的图形高度	8 位无符号	0, 255	图形像素
使用说明:				
◆ (参见 F_DISPSTRA 的使用说明)				
使用实例:				
◆ (参见 F_DISPSTRA 的使用实例)				
相关功能:				
◆ F_DISPSTRA				
◆ F_DISPTXTA				
◆ F_DISPTXTB				
◆ F_SETCURSOR				
◆ F_SETACTRECT				
◆ F_SETFILTER				
◆ F_SETINVERT				
◆ F_SETUNDERLINE				
◆ F_MODISTRCUR				

功能名称:	F_DISPXTA	功能命令代码:	63H	
功能简介:	显示通用显示信息缓冲区内的字符串, 具有自动换行功能			
输入参数个数:	6(7)	输出参数个数:	4(6)	
输入参数说明:		参数编码方式	值域	单位
[0`1]	字符串的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[2`3]	字符串信息的首地址 (在通用显示信息缓冲区内的偏移地址)	16 位无符号	0, 65535	字节
[4]	行高 (相邻两行顶端之间的间距)	7 位无符号	0, 127	-
<6..0>	显示方式			
[5]	0:普通方式; 1:复制, 2:逻辑与,	3 位无符号	0, 5	-
<2..0>	3:逻辑或, 4:逻辑非, 5:逻辑异或非			
[5]<4>	非 0:清除当前活动区域内字符串以外的所有其它已显示的内容	-	-	-
[6]	光标所在字符位的序号	8 位无符号	0, 255	-
输出参数说明:		参数编码方式	值域	单位
[0`1]	光标处字符的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	光标处字符的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4]	光标处字符的文本宽度	8 位无符号	0, 255	文本像素
[5]	光标处字符的图形高度	8 位无符号	0, 255	图形像素
使用说明:				
<ul style="list-style-type: none">◆ (参见 F_DISPSTRA 的使用说明)◆ 字符串左上角的文本横坐标为 0.◆ 当字符串中某些字符即将被活动区域的右边裁减时, 它即被显示到下一行的行首 (文本横坐标为 0).◆ 注意:换行的边界是活动区域而不是可视活动区域, 而且, 活动区域的右边界总是空出一个文本像素的宽度用以可能进行的光标显示.◆ 字符只进行一次自动换行, 即如果字符宽度大于或等于活动区域的宽度则在下一行首显示, 不会被无限地换行, 因此, 请确保活动区域的宽度大于字符串中的任意一个字符的宽度.				
使用实例:				
以下举一些字符串显示效果的例子 (假设活动区域和可视活动区域的文本宽度均为 24, 图形高度均为 80):				
<ul style="list-style-type: none">◆				
(本例中, 字符串的显示坐标为 (0, 0), 行高为 36, 其中 “,” 号后面是一个换行控制符)				
<ul style="list-style-type: none">◆				

(本例中, 字符串的显示坐标为 (0, 0), 行高为 36, 其中 “,” 号后面没有换行控制符)

相关功能:

- ◆ F_DISPXTB
- ◆ F_DISPSTRA
- ◆ F_DISPSTRB
- ◆ F_SETCURSOR
- ◆ F_SETACTRECT
- ◆ F_SETFILTER
- ◆ F_SETINVERT
- ◆ F_SETUNDERLINE
- ◆ F_MODISTRUR

功能名称:	F_DISPTXB	功能命令代码:	64H	
功能简介:	显示预置的字符串, 具有自动换行功能			
输入参数个数:	6(8)	输出参数个数:	4(6)	
输入参数说明:	参数编码方式	值域	单位	
[0`1]	字符串的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[2`3]	字符串信息的首地址(在 E ² PROM 预置数据存储区内的偏移地址)	16 位无符号	0, 65535	字节
[4]	行高(相邻两行顶端之间的间距)	7 位无符号	0, 127	-
<6..0>	显示方式			
[5]	0:普通方式; 1:复制, 2:逻辑与,	3 位无符号	0, 5	-
<2..0>	3:逻辑或, 4:逻辑非, 5:逻辑异或			
[5]<4>	非 0:清除当前活动区域内字符串以外的所有其它已显示的内容	-	-	-
[6]	光标所在字符位的序号	8 位无符号	0, 255	-
输出参数说明:	参数编码方式	值域	单位	
[0`1]	光标处字符的左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	光标处字符的左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4]	光标处字符的文本宽度	8 位无符号	0, 255	文本像素
[5]	光标处字符的的图形高度	8 位无符号	0, 255	图形像素
使用说明:				
◆ (参见 F_DISPTXA 的使用说明)				
使用实例:				
◆ (参见 F_DISPTXA 的使用实例)				
相关功能:				
◆ F_DISPTXA				
◆ F_DISPSTRA				
◆ F_DISPSTRB				
◆ F_SETCURSOR				
◆ F_SETACTRECT				
◆ F_SETFILTER				

◆ F_SETINVERT
◆ F_SETUNDERLINE
◆ F_MODISTRUCUR

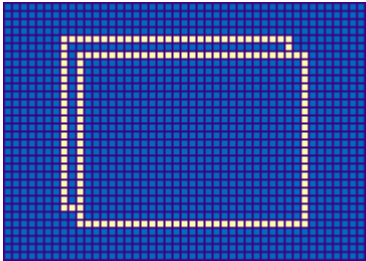
功能名称:	F_DRAWPIXEL	功能命令代码:	67H
功能简介:	显示图形像素		
输入参数个数:	3 (5)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3] 图形纵坐标	16 位补码	-1024, 1023	图形像素
[4] 图形像素的颜色	-	-	-
0:熄灭, 非 0:点亮			
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:			
◆ 输入坐标会被自动限制在值域内.			
◆ 输入坐标值是以活动区域左上角为参照的.			
◆ 显示受活动区域的限制.			
◆ 所有的绘图功能都不受过滤器, 反显方式和下划线方式的影响, 也没有逻辑显示方式.			
使用实例:			
(略)			
相关功能:			
◆ F_DISPBYTE			
◆ F_SETACTRECT			
◆ F_DRAWHLINE			
◆ F_DRAWVLINE			
◆ F_DRAWRECT			
◆ F_DRAWRECTSHD			
◆ F_DRAWSHDRECT			
◆ F_DRAWLINE			

功能名称:	F_DRAWHLINE	功能命令代码:	68H
功能简介:	绘制水平线		
输入参数个数:	4 (7)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 左端点的图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3] 左端点的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5] 水平线的图形宽度	10 位无符号	0, 1023	图形像素
[6] 线的颜色	-	-	-
0:熄灭, 非 0:点亮			
输出参数说明:	参数编码方式	值域	单位

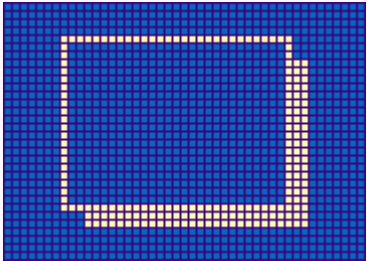
使用说明:
<ul style="list-style-type: none"> ◆ (参见 F_DRAWPIXEL 的使用说明) ◆ 输入参数[4]的第 2 位到第 7 位是无效的.
使用实例:
<ul style="list-style-type: none"> ◆ 如果左端点的图形横坐标为 6, 水平线的图形宽度为 7, 那么, 其右端点的图形横坐标为 12.
相关功能:
<ul style="list-style-type: none"> ◆ F_SETACTRECT ◆ F_DRAWPIXEL ◆ F_DRAWVLINE ◆ F_DRAWLINE

功能名称:	F_DRAWVLINE	功能命令代码:	69H
功能简介:	绘制垂直线		
输入参数个数:	4 (6)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 上端点的图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3] 上端点的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4] 垂直线的图形高度	8 位无符号	0, 255	图形像素
[5] 线的颜色	-	-	-
0:熄灭, 非 0:点亮			
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ (参见 F_DRAWPIXEL 的使用说明) 			
使用实例:			
<ul style="list-style-type: none"> ◆ 如果上端点的图形纵坐标为-6, 垂直线的图形高度为 12, 那么, 其下端点的图形纵坐标为 5. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETACTRECT ◆ F_DRAWPIXEL ◆ F_DRAWHLINE ◆ F_DRAWLINE 			

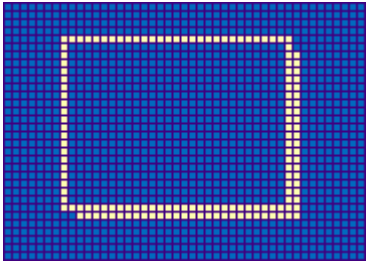
功能名称:	F_DRAWRECT	功能命令代码:	6AH
功能简介:	绘制单边矩形框		
输入参数个数:	5 (8)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 左上角的图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3] 左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5] 矩形框的图形宽度	10 位无符号	0, 1023	图形像素
[6] 矩形框的图形高度	8 位无符号	0, 255	图形像素

[7] 矩形边框的颜色				
0:熄灭, 非 0:点亮				
输出参数说明:		参数编码方式	值域	单位
-		-	-	-
使用说明:				
◆ (参见 F_DRAWHLINE 的使用说明)				
使用实例:				
◆ 可以与 F_DRAWHLINE, F_DRAWVLINE, F_DRAWPIXEL 配合实现各种矩形, 现举一例:				
				
相关功能:				
◆ F_SETACTRECT				
◆ F_DRAWHLINE				
◆ F_DRAWVLINE				
◆ F_DRAWRECTSHD				
◆ F_DRAWSHDRECT				

功能名称:	F_DRAWRECTSHD	功能命令代码:	6BH	
功能简介:	绘制单边矩形框的右下边线			
输入参数个数:	5(8)	输出参数个数:	0	
输入参数说明:	参数编码方式	值域	单位	
[0`1]	矩形框左上角的图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3]	矩形框左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	矩形框的图形宽度	10 位无符号	0, 1023	图形像素
[6]	矩形框的图形高度	8 位无符号	0, 255	图形像素
[7]	矩形边框的颜色	-	-	-
	0:熄灭, 非 0:点亮			
输出参数说明:	参数编码方式	值域	单位	
-	-	-	-	-
使用说明:				
◆ (参见 F_DRAWHLINE 的使用说明)				
◆ 可以把该功能看作是 F_DRAWRECT 功能的一部分:F_DRAWRECT 绘制完整的矩形边框, 而 F_DRAWRECTSHD 绘制的仅仅是矩形的右边线和下边线.				
使用实例:				
◆ 可以与 F_DRAWRECT 配合实现阴影更宽的矩形, 如下图所示:				



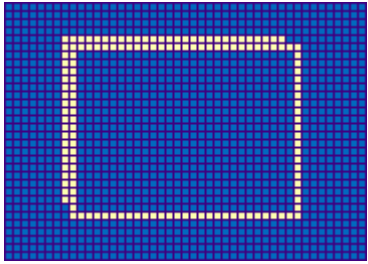
而 F_DRAWSHDRECT 只能实现阴影宽度为 1 的矩形, 如:



相关功能:

- ◆ F_SETACTRECT
- ◆ F_DRAWRECT
- ◆ F_DRAWHLINE
- ◆ F_DRAWVLINE
- ◆ F_DRAWSHDRECT

功能名称:	F_DRAWSHDRECT	功能命令代码:	6CH
功能简介:	绘制带阴影的单边矩形框		
输入参数个数:	5(8)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 矩形框左上角的图形横坐标	16 位补码	-1024, 1023	图形像素
[2`3] 矩形框左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5] 矩形框的图形宽度	10 位无符号	0, 1023	图形像素
[6] 矩形框的图形高度	8 位无符号	0, 255	图形像素
[7] 矩形边框及阴影的颜色 0:熄灭, 非 0:点亮	-	-	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ (参见 F_DRAWHLINE 的使用说明)			
◆ 阴影宽度为 1, 在矩形框的右下边, 矩形框的图形宽度和图形高度不包括阴影.			
使用实例:			
◆ (参见 F_DRAWRECTSHD 功能的使用实例)			
◆ 如果要实现其它方式的阴影, 如:			

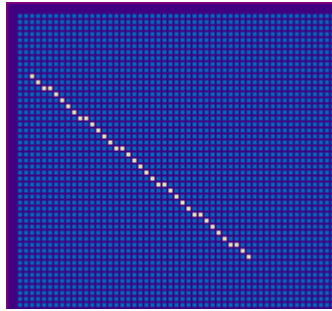


则无法用 F_DRAWSHDRECT 实现,而要分多步实现:1. 绘制左阴影垂直线(执行 F_DRAWVLINE); 2. 绘制上阴影水平线(执行 F_DRAWHLINE); 3. 绘制矩形框(执行 F_DRAWRECT).

相关功能:

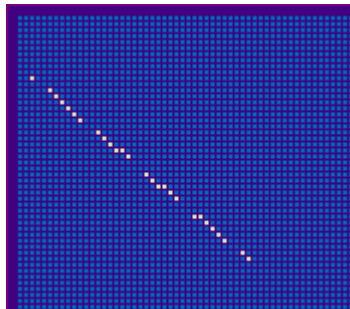
- ◆ F_SETACTRECT
- ◆ F_DRAWRECT
- ◆ F_DRAWRECTSHD
- ◆ F_DRAWHLINE
- ◆ F_DRAWVLINE

功能名称:	F_DRAWLINE		功能命令代码:	6DH	
功能简介:	绘制任意两点之间的直线段				
输入参数个数:	5 (9)		输出参数个数:	0	
输入参数说明:		参数编码方式	值域	单位	
[0`1]	线段起点的图形横坐标	16 位补码	-1024, 1023	图形像素	
[2`3]	线段起点的图形纵坐标	16 位补码	-1024, 1023	图形像素	
[4`5]	线段终点的图形横坐标	16 位补码	-1024, 1023	图形像素	
[6`7]	线段终点的图形纵坐标	16 位补码	-1024, 1023	图形像素	
[8]	线型	-	-	-	
输出参数说明:		参数编码方式	值域	单位	
-		-	-	-	
使用说明:					
◆ (参见 F_DRAWPIXEL 的使用说明)					
◆ 如果绘制的是水平线或垂直线, 推荐使用 F_DRAWHLINE 和 F_DRAWVLINE 功能, 它们的执行时间会短很多. 当然, F_DRAWLINE 的输入参数是任意两点坐标, 而 F_DRAWHLINE 和 F_DRAWVLINE 的输入参数是起始点坐标和线段长度.					
◆ 对于 F_DRAWHLINE 和 F_DRAWVLINE, 它们没有线型概念, 只有颜色概念; 使用线型则必须使用 F_DRAWLINE 功能命令.					
使用实例:					
◆ 如果活动区域设置为全屏, 输入参数设置为: [0]00H, [1]26H, [2]00H, [3]28H, [4]00H, [5]02H, [6]00H, [7]0AH, [8]FFH, 则显示结果为:					



起点坐标为 (38, 40), 终点坐标为 (2, 10).

- ◆ 如果输入参数改为: [0]00H, [1]26H, [2]00H, [3]28H, [4]00H, [5]02H, [6]00H, [7]0AH, [8]9FH, 则显示结果为:



相关功能:

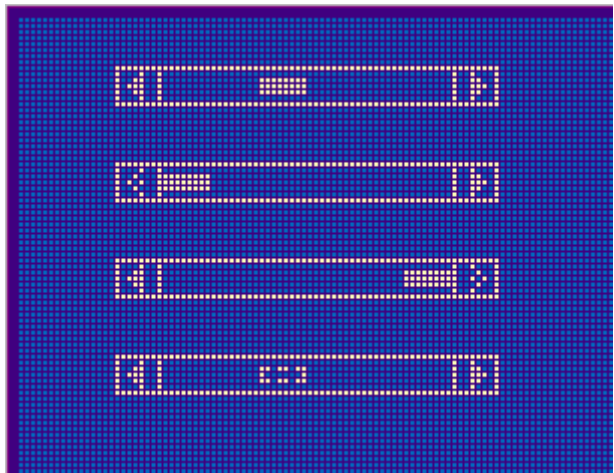
- ◆ F_SETACTRECT
- ◆ F_DRAWPIXEL
- ◆ F_DRAWHLINE
- ◆ F_DRAWVLINE

功能名称:	F_SHOWSCOLL	功能命令代码:	70H
功能简介:	显示水平滚动条		
输入参数个数:	6(8)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 左上角的文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3] 左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4] 水平滚动条的文本宽度	8 位无符号	2, 255	文本像素
[5] 滚动条最大指示值	8 位无符号	0, 255	-
[6] 滚动条当前指示值	8 位无符号	0, 255	-
[7] 焦点状态信息 (非 0:拥有焦点; 0:失去焦点)	8 位无符号	0, 255	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ 滚动条只有复制显示方式.			
◆ 滚动条最小指示值规定为 0.			
◆ 当前指示值大于最大指示值时, 被认为等于最大指示值.			

- ◆ 输入坐标会被自动限制在值域内.
- ◆ 输入坐标值是以活动区域左上角为参照的.
- ◆ 显示受活动区域的限制.
- ◆ 文本宽度小于 2 时, 水平滚动条将不会被显示出来.
- ◆ 水平滚动条的图形高度总是固定为 8.

使用实例:

- ◆ 显示四个实例, 可以对比出各参数的意义, 这四个实例的参数分别为:
第 1 个实例: [0]00H, [1]02H, [2]00H, [3]08H, [4]08H, [5]40H, [6]18H, [7]0FFH
第 2 个实例: [0]00H, [1]02H, [2]00H, [3]18H, [4]08H, [5]40H, [6]00H, [7]0FFH
第 3 个实例: [0]00H, [1]02H, [2]00H, [3]28H, [4]08H, [5]40H, [6]40H, [7]0FFH
第 4 个实例: [0]00H, [1]02H, [2]00H, [3]08H, [4]08H, [5]40H, [6]18H, [7]00H
显示在同一个页面上为:



相关功能:

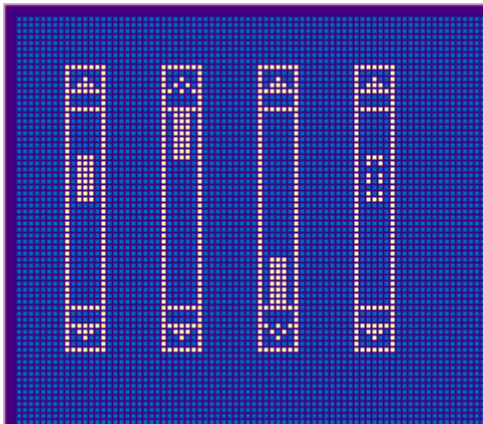
- ◆ F_SETACTRECT
- ◆ F_SHOWVSCOLL

功能名称：	F_SHOWVSCOLL	功能命令代码：	71H	
功能简介：	显示垂直滚动条			
输入参数个数：	6(8)	输出参数个数：	0	
输入参数说明：		参数编码方式	值域	单位
[0`1]	左上角的文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	左上角的图形纵坐标	16 位补码	-1024, 1023	图形像素
[4]	垂直滚动条的图形高度	8 位无符号	16, 255	图形像素
[5]	滚动条最大指示值	8 位无符号	0, 255	-
[6]	滚动条当前指示值	8 位无符号	0, 255	-
[7]	焦点状态信息 (非 0:拥有焦点； 0:失去焦点)	8 位无符号	0, 255	-
输出参数说明：		参数编码方式	值域	单位
-	-	-	-	-
使用说明：				

- ◆ (请参看 F_SHOWSCOLL 的使用说明)
- ◆ 图形高度小于 16 时, 垂直滚动条将不会被显示出来.
- ◆ 垂直滚动条的文本宽度总是固定为 1.

使用实例:

- ◆ 显示四个实例, 可以对比出各参数的意义, 这四个实例的参数分别为:
第 1 个实例: [0]00H, [1]01H, [2]00H, [3]08H, [4]30H, [5]50H, [6]18H, [7]0FFH
第 2 个实例: [0]00H, [1]03H, [2]00H, [3]08H, [4]30H, [5]50H, [6]00H, [7]0FFH
第 3 个实例: [0]00H, [1]05H, [2]00H, [3]08H, [4]30H, [5]50H, [6]50H, [7]0FFH
第 4 个实例: [0]00H, [1]07H, [2]00H, [3]08H, [4]30H, [5]50H, [6]18H, [7]00H
显示在同一个页面上为:



相关功能:

- ◆ F_SETACTRECT
- ◆ F_SHOWSCOLL

功能名称：	F_DISPUFIX16	功能命令代码：	74H	
功能简介：	无符号十六位定点数值十进制显示			
输入参数个数：	10(10)	输出参数个数：	0	
输入参数说明：		参数编码方式	值域	单位
[0`1]	起始文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	起始图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	显示有效值	16 位无符号	0, 65535	-
[6]	显示格式	-	-	-
[7]	数字-字符编码对应表编号	8 位无符号	0, 255	-
[8]	显示方式			
<2..0>	0:普通方式；1:复制，2:逻辑与，3:逻辑或，4:逻辑非，5:逻辑异或非 0:清除当前活动区域内位图以外的所有其它已显示的内容	3 位无符号	0, 5	-
[8]<4>		-	-	-
[8]<5>	对齐方式(0:左对齐；1:右对齐)	-	-	-
[8]<6>	是否调整显示位数(非 0:调整)	-	-	-

[9]	光标所在数字位的序号(权值最小的数位序号为 0)	8 位无符号	0, 255	—
输出参数说明:		参数编码方式	值域	单位
—		—	—	—
使用说明:				
<ul style="list-style-type: none"> 该功能命令必须依赖于 E²PROM 的存在, 因为“数字-字符编码对应表”信息是存放在 E²PROM 中的, 由“页面编辑器”配置生成。 “数字-字符编码对应表”规定了每个数字(0 ~ F)和必需的符号(负号, 小数点, 空白符)对应的字符编码。 输入坐标会被自动限制在值域内。 输入坐标值是以活动区域左上角为参照的。 显示受活动区域的限制。 这里的有效值并不考虑小数点的位置, 比如 34. 78 与 3. 478 和 0. 3478 的有效值都为 3478。 显示格式:<7. . 4>为有效位数, 最大值为 5;<3>为符号, 0 表示正数, 1 表示负数;<2. . 0>为小数位数, 最大值为 5;若不调整显示位数, 有效位数必须不小于小数位数, 否则无效, 不会显示数值。 参数[8]<4>的使用方法请参见 F_DISPMPB 中关于参数[7]<4>的使用实例。 左对齐方式下, 起始坐标是指左上角坐标。 右对齐方式下, (右上角的文本横坐标-1)为起始文本横坐标。 若不调整显示位数, 则显示位数即为显示格式(参数[6])中规定的有效位数;若调整显示位数, 则显示位数不受有效位数的影响, 自动根据数值确定有效位数, 如 003. 45 的显示位数为 3。 参数[9]的作用请参见 F_DISPSTRB 中参数[8]的使用说明。 				
使用实例:				
(略)				
相关功能:				
<ul style="list-style-type: none"> F_SETACTRECT F_DISPFIX16 F_MODISTRUR 				

功能名称:	F_DISPFIX16	功能命令代码:	75H	
功能简介:	十六位定点数值十进制显示			
输入参数个数:	10(10)	输出参数个数:	0	
输入参数说明:	参数编码方式	值域	单位	
[0`1]	起始文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3]	起始图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5]	显示有效值	16 位补码	-32768, 32767	-
[6]	显示格式(<3>无效)	-	-	-
[7]	数字-字符编码对应表编号	8 位无符号	0, 255	-
[8]	显示方式			
<2..0>	0: 普通方式; 1: 复制, 2: 逻辑与,	3 位无符号	0, 5	-
	3: 逻辑或, 4: 逻辑非, 5: 逻辑异或			
[8]<4>	非 0: 清除当前活动区域内位图以外的所有其它已显示的内容	-	-	-
[8]<5>	对齐方式 (0: 左对齐; 1: 右对齐)	-	-	-

[8]<6>	是否调整显示位数(非 0:调整)	-	-	-
[9]	光标所在数字位的序号(权值最小的 数位序号为 0)	8 位无符号	0, 255	-
输出参数说明:		参数编码方式	值域	单位
-	-	-	-	-
使用说明:				
◆ (参见 F_DISPUIX16 的使用说明)				
使用实例:				
(略)				
相关功能:				
◆ F_SETACTRECT				
◆ F_DISPUIX16				
◆ F_MODISTRUCUR				

功能名称:	F_VIEWSPAGE	功能命令代码:	80H
功能简介:	显示静态页面		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 页面编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
◆ 该功能命令必须有 E ² PROM 支持, 并且页面编号必须有效.			
◆ 页面信息及其编号由“页面编辑器”生成.			
◆ 该功能命令只显示页面中的静态对象.			
◆ 一般情况下, 它在同一页面只显示一次, 之后页面只更新动态部分, 可以单独对动态对象更新, 也可对动态页面(所有动态对象)进行更新.			
使用实例:			
(略)			
相关功能:			
◆ F_VIEWAPAGE			
◆ F_VIEWPAGE			
◆ F_VIEWOBJECT			

功能名称:	F_VIEWAPAGE	功能命令代码:	81H
功能简介:	显示动态页面		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 页面编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-

使用说明:
<ul style="list-style-type: none"> 该功能命令必须有 E²PROM 支持, 并且页面编号必须有效. 页面信息及其编号由“页面编辑器”生成. 该功能命令只显示页面中的动态对象.
使用实例:
(略)
相关功能:
<ul style="list-style-type: none"> F_VIEWSPAGE F_VIEWPAGE F_VIEWOBJECT

功能名称:	F_VIEWPAGE	功能命令代码:	82H
功能简介:	显示页面		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 页面编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:	<ul style="list-style-type: none"> 该功能命令必须有 E²PROM 支持, 并且页面编号必须有效. 页面信息及其编号由“页面编辑器”生成. 该功能命令显示页面中的所有对象(包括静态对象和动态对象). 该功能命令的执行效果不一定等于 F_VIEWSPAGE 和 F_VIEWAPAGE 连续执行的效果, 因为 F_VIEWSPAGE 是按顺序显示页面中的静态对象, F_VIEWAPAGE 是按顺序显示页面中的动态对象, 而 F_VIEWPAGE 是按顺序显示页面中的对象, 当存在静态对象和动态对象交替存放并且显示出现覆盖时, 两种情况不能等同. 		
使用实例:			
(略)			
相关功能:	<ul style="list-style-type: none"> F_VIEWSPAGE F_VIEWAPAGE F_VIEWOBJECT 		

功能名称:	F_VIEWOBJECT	功能命令代码:	84H
功能简介:	显示对象		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:			

- ◆ 所有与对象操作有关的功能命令必须有 E²PROM 支持, 并且对象编号必须有效.
- ◆ 对象信息及其编号由“页面编辑器”生成.
- ◆ 每个对象都有自己的活动区域, 但是, 显示对象不会影响当前的活动区域.
- ◆ 这里的对象包括所有静态对象和动态对象, 即: 所有类型对象的显示均通过执行该功能命令实现.
- ◆ 一般情况下, 我们总是用该功能命令显示动态对象, 因为静态对象常常只显示一次, 可以通过 F_VIEWSPAGE 命令批量执行.

使用实例:

(略)

相关功能:

- ◆ F_VIEWSPAGE
- ◆ F_VIEWAPAGE
- ◆ F_VIEWPAGE

功能名称:	F_SETFOCUSED	功能命令代码:	88H
功能简介:	设置对象的焦点信息		
输入参数个数:	2(3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 焦点信息 (非 0:拥有焦点; 0:失去焦点)	-	-	-
[1`2] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	<ul style="list-style-type: none"> ◆ 仅对能够拥有焦点的对象起作用. ◆ 设置之后需要重新显示该对象(执行 F_VIEWOBJECT 命令)才能更新显示内容. 		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_GETFOCUSED ◆ F_VIEWOBJECT 		

功能名称:	F_GETFOCUSED	功能命令代码:	89H
功能简介:	获得对象的焦点信息		
输入参数个数:	1(2)	输出参数个数:	1(1)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
[0] 对象焦点信息 (非 0:拥有焦点; 0:失去焦点)	-	-	-
使用说明:	<ul style="list-style-type: none"> ◆ 对象必须能够拥有焦点, 否则所得结果无意义. 		

使用实例:
(略)
相关功能:
◆ F_SETFOCUSED

功能名称:	F_GETOBJITEMS	功能命令代码:	8CH
功能简介:	获得单选框, 复选框, 列表框的选项数目或数值编辑器的有效位数		
输入参数个数:	1 (2)	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
[0] 选项数目或编辑器的有效位数	8 位无符号	0, 255	-
使用说明:	◆ 仅对单选框, 复选框, 列表框和数值编辑器有效, 若针对其它对象, 所得结果无意义. ◆ 该功能命令常与 F_SETOBJCUR, F_GETOBJCUR 命令配合使用.		
使用实例:	(略)		
相关功能:	◆ F_SETOBJCUR ◆ F_GETOBJCUR ◆ F_INCOBJCUR ◆ F_DECOBJCUR		

功能名称:	F_SETOBJCUR	功能命令代码:	8DH
功能简介:	设置单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号		
输入参数个数:	2 (3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 选中项序号或光标位序号	8 位无符号	0, 255	-
[1`2] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	◆ 仅对单选框, 复选框, 列表框和数值编辑器有效. ◆ 最小的序号为 0. ◆ 若序号设置值大于或等于选项数目(或有效位数), 则会被自动取余. ◆ 编辑器中, 权值最小的位序号为 0. ◆ 需要重新显示对象才能更新显示内容.		
使用实例:	◆ 如果某编辑器的有效位数为 4, 光标位序号的设置值为 6, 则, 实际设置值为 2.		
相关功能:	◆ F_GETOBJCUR		

◆ F_GETOBJITEMS
◆ F_INCOBJCUR
◆ F_DECOBJCUR
◆ F_VIEWOBJECT

功能名称:	F_GETOBJCUR	功能命令代码:	8EH
功能简介:	获得单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号		
输入参数个数:	1 (2)	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0] 选中项序号或光标位序号	8 位无符号	0, 255	—
使用说明:	◆ 仅对单选框, 复选框, 列表框和数值编辑器有效, 若针对其它对象, 所得结果无意义.		
使用实例:	(略)		
相关功能:	◆ F_SETOBJCUR ◆ F_INCOBJCUR ◆ F_DECOBJCUR		

功能名称:	F_INCOBJCUR	功能命令代码:	8FH
功能简介:	循环递增单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:	◆ 仅对单选框, 复选框, 列表框和数值编辑器有效. ◆ 编辑器中, 权值最小的位序号为 0. ◆ 所谓循环递增, 是指: 若序号递增后等于选项数目 (或编辑器有效位数), 则归 0. ◆ 需要重新显示对象才能更新显示内容.		
使用实例:	◆ 如果某编辑器的有效位数为 4, 当前光标位序号的设置值为 2, 则执行该功能命令后的实际光标位序号为 3. ◆ 如果某编辑器的有效位数为 4, 当前光标位序号的设置值为 3, 则执行该功能命令后的实际光标位序号为 0.		
相关功能:	◆ F_SETOBJCUR ◆ F_GETOBJCUR		

- ◆ F_DECOBJCUR
- ◆ F_VIEWOBJECT

功能名称:	F_DECOBJCUR	功能命令代码:	90H
功能简介:	循环递减单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
<ul style="list-style-type: none"> ◆ 仅对单选框, 复选框, 列表框和数值编辑器有效. ◆ 编辑器中, 权值最小的位序号为 0. ◆ 所谓循环递减, 是指: 若当前序号等于 0, 则设置成最大序号值 (等于选项数目-1). ◆ 需要重新显示对象才能更新显示内容. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 如果某编辑器的有效位数为 4, 当前光标位序号的设置值为 2, 则执行该功能命令后的实际光标位序号为 1. ◆ 如果某编辑器的有效位数为 4, 当前光标位序号的设置值为 0, 则执行该功能命令后的实际光标位序号为 3. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETOBJCUR ◆ F_GETOBJCUR ◆ F_INCOBJCUR ◆ F_VIEWOBJECT 			

功能名称:	F_SETCHECK	功能命令代码:	92H
功能简介:	设置复选框所有选项的状态		
输入参数个数:	3 (3+N)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0..N-1] 状态信息	—	—	—
[N] 状态信息的字节数 (N)	8 位无符号	0, 255	—
[N+1`N+2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
<ul style="list-style-type: none"> ◆ 仅对复选框对象有效. ◆ 需要重新显示对象才能更新显示内容. ◆ 状态信息的格式为每个选项用一位数据表示, 第 0 个字节的第 0 位为第 0 项状态, 第 0 个字节的第 			

1 位为第 1 项状态, 第 1 个字节的第 0 位为第 8 项状态, 以此类推.
使用实例:
(略)
相关功能:
◆ F_GETCHECK
◆ F_VIEWOBJECT

功能名称:	F_GETCHECK	功能命令代码:	93H
功能简介:	获得复选框所有选项的状态		
输入参数个数:	1 (2)	输出参数个数:	2 (1+N)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0] 状态信息的字节数 (N)	8 位无符号	0, 255	—
[1..N] 状态信息	—	—	—
使用说明:	◆ 仅对复选框对象有效, 若针对其它对象, 所得结果无意义.		
使用实例:	(略)		
相关功能:	◆ F_SETCHECK		

功能名称:	F_ALTCHKISTA	功能命令代码:	94H
功能简介:	切换复选框某项的状态		
输入参数个数:	2 (3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 切换项序号	8 位无符号	0, 255	—
[1`2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:	◆ 仅对复选框对象有效. ◆ 需要重新显示对象才能更新显示内容. ◆ 切换项序号的有效性必须自行把握, 可以通过 F_GETOBJITEMS 获得选项数目.		
使用实例:	(略)		
相关功能:	◆ F_GETCHKISTA ◆ F_ALTCHKESTA ◆ F_GETCHKESTA ◆ F_GETOBJITEMS		

◆ F_VIEWOBJECT

功能名称:	F_GETCHKISTA	功能命令代码:	95H
功能简介:	获得复选框某项的状态		
输入参数个数:	2(3)	输出参数个数:	1(1)
输入参数说明:	参数编码方式	值域	单位
[0] 选项序号	8 位无符号	0, 255	—
[1`2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0] 选项状态 (非 0: 表示状态 “1” ; 0: 表示状态 “0”)	—	—	—
使用说明:	<ul style="list-style-type: none"> ◆ 仅对复选框对象有效, 否则所得结果无意义. ◆ 选项序号的有效性必须自行把握, 可以通过 F_GETOBJITEMS 获得选项数目. 		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_SETCHKISTA ◆ F_GETCKESTA ◆ F_SETCKESTA ◆ F_GETOBJITEMS 		

功能名称:	F_ALTCKESTA	功能命令代码:	96H
功能简介:	切换复选框选中项的状态		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:	<ul style="list-style-type: none"> ◆ 仅对复选框对象有效. ◆ 需要重新显示对象才能更新显示内容. ◆ 可以执行 F_GETOBJCUR 获得当前选中项序号. 		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none"> ◆ F_GETCKESTA ◆ F_GETOBJCUR ◆ F_SETCHKISTA ◆ F_GETCHKISTA ◆ F_GETOBJITEMS 		

◆ F_VIEWOBJECT

功能名称:	F_GETCKESTA	功能命令代码:	97H
功能简介:	获得复选框选中项的状态		
输入参数个数:	1 (2)	输出参数个数:	1 (1)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0] 选项状态 (非 0: 表示状态 “1” ; 0: 表示状态 “0”)	—	—	—
使用说明:			
◆ 仅对复选框对象有效, 否则所得结果无意义.			
◆ 可以执行 F_GETOBJCUR 获得当前选中项序号.			
使用实例:			
(略)			
相关功能:			
◆ F_SETCKESTA			
◆ F_GETOBJCUR			
◆ F_GETCHKISTA			
◆ F_SETCHKISTA			

功能名称:	F_INCLSTIS_V	功能命令代码:	9AH
功能简介:	限制递增列表框选中项行号		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:			
◆ 仅对列表框对象有效.			
◆ 所谓限制递增, 是指: 若当前行号等于该列的最大行号, 则不再递增.			
◆ 需要重新显示对象才能更新显示内容.			
使用实例:			
◆ 对于相同的前提, 某列表框的选项数目为 7, 分 3 列显示:			
选项 0 选项 1 选项 2			
选项 3 选项 4 选项 5			
选项 6			
若当前选中项序号为 1, 则执行该功能命令后选中项序号为 4;			
若当前选中项序号为 4, 则执行该功能命令后不改变选中项的序号;			
若当前选中项的序号为 3, 则执行该功能命令后选中项序号为 6.			
相关功能:			

◆ F_DECLSTIS_V
◆ F_VIEWOBJECT
◆ F_GETOBJCUR

功能名称:	F_DECLSTIS_V	功能命令代码:	9BH
功能简介:	限制递减列表框选中项行号		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:			
◆ 仅对列表框对象有效.			
◆ 所谓限制递减, 是指: 若当前行号等于 0, 则不再递减.			
◆ 需要重新显示对象才能更新显示内容.			
使用实例:			
◆ 对于相同的前提, 某列表框的选项数目为 7, 分 3 列显示:			
选项 0 选项 1 选项 2			
选项 3 选项 4 选项 5			
选项 6			
若当前选中项序号为 1, 则执行该功能命令后选中项序号不变;			
若当前选中项序号为 4, 则执行该功能命令后选中项序号为 1.			
相关功能:			
◆ F_INCLSTIS_V			
◆ F_VIEWOBJECT			
◆ F_GETOBJCUR			

功能名称:	F_SETSCVALUE	功能命令代码:	9EH
功能简介:	设置滚动条当前值和最大值		
输入参数个数:	3 (4)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 当前值	8 位无符号	0, 255	-
[1] 最大值	8 位无符号	0, 255	-
[2`3] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
- -	-	-	-
使用说明:			
◆ 仅对滚动条对象有效.			
◆ 需要重新显示对象才能更新显示内容.			
◆ 若当前值大于最大值, 则将最大值调整为当前值.			
◆ 最小值为 0.			

使用实例:
(略)
相关功能:
◆ F_GETSCVALUE
◆ F_VIEWOBJECT

功能名称:	F_GETSCVALUE	功能命令代码:	9FH
功能简介:	获得滚动条当前值和最大值		
输入参数个数:	1(2)	输出参数个数:	2(2)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0] 当前值	8 位无符号	0, 255	—
[1] 最大值	8 位无符号	0, 255	—
使用说明:	◆ 仅对滚动条对象有效, 否则所得结果无意义.		
使用实例:	(略)		
相关功能:	◆ F_SETSCVALUE		

功能名称:	F_SETVALFMT	功能命令代码:	A2H
功能简介:	设置动态数值框的显示格式		
输入参数个数:	2(3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 显示格式	—	—	—
[1`2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:	◆ 仅对动态数值框对象有效. ◆ 需要重新显示对象才能更新显示内容. ◆ 数值框的种类有三种: 普通, 指定符号的标准十进制, 标准十进制. ◆ 普通数值框的显示格式: <7..4> 为有效位数, 最大值为 15; <3> 为符号, 0 表示正数, 1 表示负数; <2..0> 为小数位数, 最大值为 7; 有效位数必须不小于小数位数, 否则无效, 不会显示数值. ◆ 指定符号的标准十进制数值框的显示格式: <7..4> 为有效位数, 最大值为 5; <3> 为符号, 0 表示正数, 1 表示负数; <2..0> 为小数位数, 最大值为 5; 若不调整显示位数, 有效位数必须不小于小数位数, 否则无效, 不会显示数值. ◆ 标准十进制数值框的显示格式: <7..4> 为有效位数, 最大值为 5; <2..0> 为小数位数, 最大值为 5; 若不调整显示位数, 有效位数必须不小于小数位数, 否则无效, 不会显示数值.		
使用实例:			

(略)
相关功能:
◆ F_SETVALUE
◆ F_VIEWOBJECT

功能名称:	F_SETVALUE	功能命令代码:	A3H
功能简介:	设置动态数值框的显示数据		
输入参数个数:	2 (2+N)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0..N-1] 显示数据	-	-	-
[N`N+1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	<ul style="list-style-type: none">◆ (参见 F_SETVALFMT 的使用说明)◆ 普通动态数值框的显示数据最多为 8 个字节(可根据有效位数决定显示数据的字节数, 每个字节可包含两位有效数字), 高字节先送.◆ 指定符号的标准十进制动态数值框的显示数据为 2 个字节(16 位无符号数值), 高字节先送.◆ 标准十进制动态数值框的显示数据为 2 个字节(16 位补码数值), 高字节先送.◆ 显示数值超过 65535 的必须采用普通动态数值框显示, 且只按每个半字节对应显示, 如果打算以十进制显示这个数值, 必须事先将其转换成压缩 BCD 码格式.		
使用实例:	<ul style="list-style-type: none">◆ 若动态数值框为普通类型, 显示格式为 74H(7 位有效数字, 4 位小数, 正号), 不调整显示位数, 显示数据为 [0..3]=00A5E37DH, 则显示结果为: 0A5. E37D 若想以十进制方式显示值为 00A5E37DH 的数, 则输入参数应为 [0..3]=10871677H, 显示结果为: 087. 1677 因为有效位数为 7, 所以最高位“1”被截去了.		
相关功能:	<ul style="list-style-type: none">◆ F_SETVALFMT◆ F_VIEWOBJECT		

功能名称:	F_SETSTDELMT	功能命令代码:	A6H
功能简介:	设置标准十进制定点数值编辑器的数值极限, 并重新调整编辑数值		
输入参数个数:	3 (6)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 数值下限	16 位补码	-32768, 32767	-
[2`3] 数值上限	16 位补码	-32768, 32767	-
[4`5] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位

使用说明:
<ul style="list-style-type: none">◆ 仅对标准十进制定点数值编辑器对象有效.◆ 需要重新显示对象才能更新显示内容.◆ 光标是否显示以及是否闪烁, 请自行控制.◆ 标准十进制定点数值编辑器的最大编辑范围是 (-32768, 32767).◆ 数值上限须不小于数值下限, 否则, 数值上限和下限会被分别改成 32767 和 -32768.◆ 一般情况下, 在使用数值编辑器之前, 需先设置数值极限, 再设置编辑数值的初值.◆ 标准十进制定点数值编辑器仅支持定点定长方式, 若编辑长度小于 5, 请自行控制数值上下限的值.
使用实例:
<ul style="list-style-type: none">◆ 如原编辑数值为 300, 现设置数值下限和上限分别为 400 和 800, 则编辑数值会被调整为 400.◆ 若原编辑数值为 300, 现设置数值下限和上限分别为 100 和 200, 则编辑数值会被调整为 200.
相关功能:
<ul style="list-style-type: none">◆ F_SETSTDEVAL◆ F_GETSTDEVAL◆ F_VIEWOBJECT◆ F_SETCURSOR◆ F_SETCURGLIT

功能名称:	F_SETSTDEVAL	功能命令代码:	A7H
功能简介:	设置标准十进制定点数值编辑器的编辑数值.		
输入参数个数:	2 (4)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 编辑数值	16 位补码	-32768, 32767	-
[2`3] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	◆ (请参看 F_SETSTDELMT 的使用说明).		
使用实例:	(略)		
相关功能:	<ul style="list-style-type: none">◆ F_SETSTDELMT◆ F_GETSTDEVAL◆ F_VIEWOBJECT◆ F_SETCURSOR◆ F_SETCURGLIT		

功能名称:	F_GETSTDEVAL	功能命令代码:	A8H
功能简介:	获得标准十进制定点数值编辑器的编辑数值和数值极限		

输入参数个数:	1 (2)	输出参数个数:	3 (6)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
[0`1] 编辑数值	16 位补码	-32768, 32767	—
[2`3] 数值上限	16 位补码	-32768, 32767	—
[4`5] 数值下限	16 位补码	-32768, 32767	—
使用说明:			
<ul style="list-style-type: none"> ◆ 仅对标准十进制定点数值编辑器对象有效, 否则所得结果无意义. ◆ 一般情况下, 只读取编辑数值即可, 这样可以节省时间. 			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETSTDELMT ◆ F_SETSTDEVAL 			

功能名称:	F_SETCURVALUE	功能命令代码:	A9H
功能简介:	设置标准数值编辑器光标处数值		
输入参数个数:	2 (3)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 光标处一位数字	8 位无符号	0, 9	—
[1`2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:			
<ul style="list-style-type: none"> ◆ (请参看 F_SETSTDELMT 的使用说明). ◆ 参数[0]的值域为 (0, 9), 超出这个值域将被 10 取余, 例如设置值 27 将被改成 7. ◆ 如果设置后的数值超出上下限, 则该设置无效, 恢复原值. ◆ 改变光标位置请执行 F_SETOBJCUR, F_INCOBJCUR, F_DECOBJCUR 等功能命令. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 如原值为 3465, 上下限分别为 2790 和 3800, 光标位序号为 2, 即: 光标位数字为 4. 若欲将光标位数字改为 9, 则新值 3965 将超出上限, 因此设置无效, 保持原值. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETSTDELMT ◆ F_GETSTDEVAL ◆ F_INCCURVALUE ◆ F_DECCURVALUE ◆ F_ALTSTDVDESIGN ◆ F_VIEWOBJECT ◆ F_SETCURSOR ◆ F_SETCURGLIT ◆ F_SETOBJCUR 			

- ◆ F_INCOBJCUR
- ◆ F_DECOBJCUR

功能名称:	F_INCCURVALUE	功能命令代码:	AAH
功能简介:	标准数值编辑器光标处数值增一并且向前进位		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:	◆ (请参看 F_SETCURVALUE 的使用说明).		
使用实例:	◆ 如原值为 3795, 上下限分别为 2000 和 3850, 光标位序号为 1, 即: 光标位数字为 9. 若欲将光标位数字增一, 则新值为 3805.		
相关功能:	◆ F_SETSTDELMT ◆ F_GETSTDEVAL ◆ F_DECCURVALUE ◆ F_ALTSTDVESIGN ◆ F_VIEWOBJECT ◆ F_SETCURSOR ◆ F_SETCURGLIT ◆ F_SETOBJCUR ◆ F_INCOBJCUR ◆ F_DECOBJCUR		

功能名称:	F_DECCURVALUE	功能命令代码:	ABH
功能简介:	标准数值编辑器光标处数值减一并且向前借位		
输入参数个数:	1(2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
— —	—	—	—
使用说明:	◆ (请参看 F_SETCURVALUE 的使用说明).		
使用实例:	(略)		
相关功能:	◆ F_SETSTDELMT ◆ F_GETSTDEVAL		

- ◆ F_INCCURVALUE
- ◆ F_ALTSTDVESIGN
- ◆ F_VIEWOBJECT
- ◆ F_SETCURSOR
- ◆ F_SETCURGLIT
- ◆ F_SETOBJCUR
- ◆ F_INCOBJCUR
- ◆ F_DECOBJCUR

功能名称:	F_ALTSTDVESIGN	功能命令代码:	ACH
功能简介:	标准编辑器数值符号取反		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ (请参看 F_SETSTDELMT 的使用说明). ◆ 所谓数值符号取反, 就是求原值的负数 ◆ 如果设置后的数值超出上下限, 则该设置无效, 恢复原值. 			
使用实例:			
(略)			
相关功能:			
<ul style="list-style-type: none"> ◆ F_SETSTDELMT ◆ F_GETSTDEVAL ◆ F_INCCURVALUE ◆ F_DECCURVALUE ◆ F_VIEWOBJECT ◆ F_SETCURSOR ◆ F_SETCURGLIT 			

功能名称:	F_GETATXT_LEN	功能命令代码:	BOH
功能简介:	获得动态文本框的长度和最大长度		
输入参数个数:	1 (2)	输出参数个数:	2 (2)
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
[0] 字符串长度	8 位无符号	0, 255	字符
[1] 字符串最大长度	8 位无符号	0, 255	字符
使用说明:			
◆ 仅对动态文本框对象有效, 否则所得结果无意义.			

- ◆ 字符串长度的设置是在更改字符串内容时自动进行的。

使用实例：

(略)

相关功能：

- ◆ F_ATXTCLR
- ◆ F_ATXTAPPEND
- ◆ F_ATXTINSERT
- ◆ F_ATXTDELETE

功能名称：	F_SETATXT_XY	功能命令代码：	B1H
功能简介：	设置动态文本框显示内容的起始坐标		
输入参数个数：	3(6)	输出参数个数：	0
输入参数说明：	参数编码方式	值域	单位
[0`1] 左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3] 左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
[4`5] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明：	参数编码方式	值域	单位
-	-	-	-
使用说明：			
<ul style="list-style-type: none"> ◆ 仅对动态文本框对象有效。 ◆ 需要重新显示对象才能更新显示内容。 ◆ 输入参数的坐标不是对象的坐标, 而是字符串在对象所在活动区域内的坐标。 ◆ 如果动态文本框为换行方式, 则横坐标无效, 但仍然需要输入两个字节的无关数值。 			
使用实例：			
(略)			
相关功能：			
<ul style="list-style-type: none"> ◆ F_GETATXT_XY ◆ F_VIEWOBJECT 			

功能名称：	F_GETATXT_XY	功能命令代码：	B2H
功能简介：	获得动态文本框显示内容的起始坐标		
输入参数个数：	1(2)	输出参数个数：	2(4)
输入参数说明：	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明：	参数编码方式	值域	单位
[0`1] 左上角文本横坐标	16 位补码	-1024, 1023	文本像素
[2`3] 左上角图形纵坐标	16 位补码	-1024, 1023	图形像素
使用说明：			
<ul style="list-style-type: none"> ◆ 仅对动态文本框对象有效, 否则所得结果无意义。 ◆ 所得参数的坐标不是对象的坐标, 而是字符串在对象所在活动区域内的坐标。 ◆ 如果动态文本框为换行方式, 则横坐标无效。 			

使用实例:
(略)
相关功能:
◆ F_SETATXT_XY

功能名称:	F_ATXTCLR	功能命令代码:	B3H
功能简介:	清动态文本框的字符串内容		
输入参数个数:	1 (2)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0`1] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
◆ 仅对动态文本框对象有效.			
◆ 需要重新显示对象才能更新显示内容.			
使用实例:			
(略)			
相关功能:			
◆ F_GETATXT_LEN			
◆ F_VIEWOBJECT			
◆ F_ATXTAPPEND			
◆ F_ATXTINSERT			
◆ F_ATXTDELETE			

功能名称:	F_ATXTAPPEND	功能命令代码:	B4H
功能简介:	在动态文本框的末尾添加文本		
输入参数个数:	3 (3+2N)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0.. 2N-1] 添加的字符编码序列	—	—	—
[2N] 添加字符数	8 位补码	0, 6	字符
[2N+1` 2N+2] 对象编码	16 位无符号	0, FFFFH	—
输出参数说明:	参数编码方式	值域	单位
—	—	—	—
使用说明:			
◆ 仅对动态文本框对象有效.			
◆ 需要重新显示对象才能更新显示内容.			
◆ 每次最多添加 6 个字符.			
◆ 超出文本最大长度的字符被裁减.			
◆ 最大长度在“页面编辑器”中设置.			

使用实例:
<ul style="list-style-type: none"> ◆ 若文本最大长度为 10, 原字符串为“杭州”, 现欲添加“士兰”两个标准字符, 则输入参数 [0..3]=CABFCOBCH, [4]=2, 其中: “士”字的编码为 CABFH, “兰”字的编码为 COBCH. 添加后的字符串为“杭州士兰”. ◆ 上例中, 若文本最大长度改为 3, 则添加“士兰”两个字符后, 字符串为“杭州士”.
相关功能:
<ul style="list-style-type: none"> ◆ F_GETATXT_LEN ◆ F_VIEWOBJECT ◆ F_ATXTCLR ◆ F_ATXTINSERT ◆ F_ATXTDELETE

功能名称:	F_ATXTINSERT	功能命令代码:	B5H
功能简介:	在动态文本框中插入文本		
输入参数个数:	4 (4+2N)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0.. 2N-1]	插入的字符编码序列	-	-
[2N]	插入字符数	8 位补码	0, 6
[2N+1]	插入点序号	8 位无符号	0, 255
[2N+2.. 2N+3]	对象编码	16 位无符号	0, FFFFH
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:	<ul style="list-style-type: none"> ◆ 仅对动态文本框对象有效. ◆ 需要重新显示对象才能更新显示内容. ◆ 每次最多插入 6 个字符. ◆ 超出文本最大长度的字符被裁减, 与 F_ATXTAPPEND 不同, 这里被裁减的字符可能是原来已有的字符. ◆ 若插入点序号大于或等于现有字符串的长度, 则效果与添加文本相同. 		
使用实例:	<ul style="list-style-type: none"> ◆ 若文本最大长度为 10, 原字符串为“杭州”, 现欲在“杭”字之前插入“士兰”两个标准字符, 则输入参数 [0..3]=CABFCOBCH, [4]=2, [5]=0, 其中: “士”字的编码为 CABFH, “兰”字的编码为 COBCH. 添加后的字符串为“士兰杭州”. ◆ 上例中, 若文本最大长度改为 3, 则插入“士兰”两个字符后, 字符串为“士兰杭”. 		
相关功能:	<ul style="list-style-type: none"> ◆ F_GETATXT_LEN ◆ F_VIEWOBJECT ◆ F_ATXTCLR ◆ F_ATXTAPPEND ◆ F_ATXTDELETE 		

功能名称:	F_ATXTDELETE	功能命令代码:	B6H
功能简介:	在动态文本框中删除文本		
输入参数个数:	3(4)	输出参数个数:	0
输入参数说明:	参数编码方式	值域	单位
[0] 删除字符数	8 位补码	0, 255	字符
[1] 删除点序号	8 位无符号	0, 255	-
[2`3] 对象编码	16 位无符号	0, FFFFH	-
输出参数说明:	参数编码方式	值域	单位
-	-	-	-
使用说明:			
<ul style="list-style-type: none"> ◆ 仅对动态文本框对象有效. ◆ 需要重新显示对象才能更新显示内容. 			
使用实例:			
<ul style="list-style-type: none"> ◆ 若原字符串为“ABCDEF”, 现欲删除“BCD”三个字符, 则输入参数[0]=3, [1]=1, 删除后的字符串为“AEF”. ◆ 若原字符串为“ABCDEF”, 输入参数[0]=8, [1]=2, 则执行之后字符串为“AB”. 			
相关功能:			
<ul style="list-style-type: none"> ◆ F_GETATXT_LEN ◆ F_VIEWOBJECT ◆ F_ATXTCLR ◆ F_ATXTAPPEND ◆ F_ATXTINSERT 			

四、对象类型汇总

通用图形点阵液晶显示控制器的对象类型有 11 个：

1. OBJ_STXT 静态文本
2. OBJ_BMP 静态位图
3. OBJ_BLOCK 静态矩形填充区块
4. OBJ_SHAPE 静态图形(组合图形, 由线段构成)
5. OBJ_ATXT 动态文本
6. OBJ_VALUE 动态数值显示
7. OBJ_RADIO 单选框
8. OBJ_CHECK 复选框
9. OBJ_LIST 通用列表框
10. OBJ_SCROLL 滚动条
11. OBJ_STDVEDIT 标准十进制定点数值编辑器

五、对象操作方法汇总

1. 静态文本		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象

2. 静态位图		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象

3. 静态矩形填充区块		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象

4. 静态图形		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象

5. 动态文本		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_GETATXT_LEN	B0H	获得动态文本框的长度和最大长度
F_SETATXT_XY	B1H	设置动态文本框显示内容的坐标

F_GETATXT_XY	B2H	获得动态文本框显示内容的坐标
F_ATXTCLR	B3H	清动态文本框的字符串内容
F_ATXTAPPEND	B4H	在动态文本框的末尾添加文本
F_ATXTINSERT	B5H	在动态文本框中插入文本
F_ATXTDELETE	B6H	在动态文本框中删除字符串

6. 动态数值显示		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETVALFMT	A2H	设置动态数值框的显示格式
F_SETAVALUE	A3H	设置动态数值框的显示数据

7. 单选框		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJITEMS	8CH	获得选项数目
F_SETOBJCUR	8DH	设置选中项序号
F_GETOBJCUR	8EH	获得选中项序号
F_INCOBJCUR	8FH	循环递增选中项序号
F_DECOBJCUR	90H	循环递减选中项序号

8. 复选框		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJITEMS	8CH	获得选项数目
F_SETOBJCUR	8DH	设置选中项序号
F_GETOBJCUR	8EH	获得选中项序号
F_INCOBJCUR	8FH	循环递增选中项序号
F_DECOBJCUR	90H	循环递减选中项序号
F_SETCHECK	92H	设置复选框所有选项状态
F_GETCHECK	93H	获得复选框所有选项状态
F_ALTCHKISTA	94H	切换复选框某项状态
F_GETCHKISTA	95H	获得复选框某项状态
F_ALTCHKESTA	96H	切换复选框选中项状态
F_GETCHKESTA	97H	获得复选框选中项状态

9. 通用列表框		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJITEMS	8CH	获得选项数目
F_SETOBJCUR	8DH	设置选中项序号
F_GETOBJCUR	8EH	获得选中项序号
F_INCOBJCUR	8FH	循环递增选中项序号
F_DECOBJCUR	90H	循环递减选中项序号
F_INCLSTIS_V	9AH	限制递增列表框选中项行号
F_DECLSTIS_V	9BH	限制递减列表框选中项行号

10. 滚动条		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_SETSCVALUE	9EH	设置滚动条当前值和最大值
F_GETSCVALUE	9FH	获得滚动条当前值和最大值

11. 标准十进制定点数值编辑器		
功能名称	命令 代码	功能简介
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJITEMS	8CH	获得有效位数
F_SETOBJCUR	8DH	设置光标位序号
F_GETOBJCUR	8EH	获得光标位序号
F_INCOBJCUR	8FH	循环递增光标位序号
F_DECOBJCUR	90H	循环递减光标位序号
F_SETSTDELMT	A6H	设置标准十进制定点数值编辑器的数值极限, 并重新调整编辑数值
F_SETSTDEVAL	A7H	设置标准十进制定点数值编辑器的数值
F_GETSTDEVAL	A8H	获得标准十进制定点数值编辑器的数值和极限值
F_SETCURVALUE	A9H	设置标准数值编辑器光标处数值
F_INCCURVALUE	AAH	标准数值编辑器光标处数值增一并且向前进位
F_DECCURVALUE	ABH	标准数值编辑器光标处数值减一并且向前借位
F_ALTSTDVESIGN	ACH	标准编辑器数值符号取反

六、过滤器/填充模板

相同编号的过滤器和填充模板具有相同的点阵图案，如下表所示：

L H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

七、功能索引

功能名称	命令 代码	功能简介
F_SETBASIC	00H	设置液晶的基本参数
F_GETBASIC	01H	获取液晶基本参数的设置值
F_SETOSCFREQ	02H	设置外接晶振频率
F_GETOSCFREQ	03H	获取外接晶振频率的设置值
F_SETDISP	04H	设置液晶显示开关
F_GETLSW	05H	获取液晶状态字
F_SETPAGEINFO	08H	设置页面信息, 包括显示页号和操作页号
F_GETPAGEINFO	09H	获取页面信息, 包括显示页号和操作页号
F_SETACTRECT	0CH	设置活动区域
F_GETACTRECT	0DH	获取活动区域
F_SETACTFULL	0EH	设置整个页面区域为活动区域
F_SETFILTER	10H	设置过滤器编号
F_GETFILTER	11H	获取过滤器编号
F_SETINVERT	12H	设置反显方式
F_GETINVERT	13H	获取反显方式
F_SETUNDERLINE	14H	设置下划线方式
F_GETUNDERLINE	15H	获取下划线方式
F_SETCURSOR	18H	光标显示开关
F_SETCURGLIT	19H	光标闪烁开关
F_MOVECURSOR	1AH	移动光标
F_RESIZECURSOR	1BH	改变光标尺寸
F_MODISTRCUR	1CH	将字符串显示后产生的光标信息赋给光标
F_SETBDPTR	24H	设置通用显示信息缓冲区数据指针
F_GETBDPTR	25H	获取通用显示信息缓冲区的数据指针
F_WRBUFA	28H	往通用显示信息缓冲区写一个数据, 数据指针循环增一
F_WRBUFB	29H	往通用显示信息缓冲区写 N 个数据, 数据指针循环增加
F_WRBUFC	2AH	往通用显示信息缓冲区写 16 个数据, 数据指针循环增加
F_WRBUFD	2BH	往通用显示信息缓冲区写一个数据, 但不改变数据指针
F_RDBUFB	2CH	从通用显示信息缓冲区读 N 个数据, 数据指针循环增加
F_RDBUFD	2DH	从通用显示信息缓冲区读 N 个数据, 但不改变数据指针
F_SETFDPTR	34H	设置自由存取区数据指针
F_GETFDPTR	35H	获取自由存取区的数据指针
F_WRFREEA	38H	往自由存取区写一个数据, 数据指针循环增一
F_WRFREEB	39H	往自由存取区写 N 个数据, 数据指针循环增加
F_WRFREEC	3AH	往自由存取区写 16 个数据, 数据指针循环增加
F_WRFREED	3BH	往自由存取区写一个数据, 但不改变数据指针
F_RDFREEB	3CH	从自由存取区读 N 个数据, 数据指针循环增加
F_RDFREED	3DH	从自由存取区读 N 个数据, 但不改变数据指针

F_EEPWREN	42H	设置可编程自由存取区的写保护开关
F_SETEDPTR	44H	设置可编程自由存取区数据指针
F_GETEDPTR	45H	获取可编程自由存取区的数据指针
F_WREEPFA	48H	往可编程自由存取区写一个数据, 数据指针循环增一
F_WREEPFB	49H	往可编程自由存取区写 N 个数据, 数据指针循环增加
F_WREEPFC	4AH	往可编程自由存取区写 16 个数据, 数据指针循环增加
F_WREEPFD	4BH	往可编程自由存取区写一个数据, 但不改变数据指针
F_RDEEPFB	4CH	从可编程自由存取区读 N 个数据, 数据指针循环增加
F_RDEEPFD	4DH	从可编程自由存取区读 N 个数据, 但不改变数据指针
F_COPYPAGE	50H	复制整个页面
F_SAVEACTRECT	51H	保存活动区域的内容到其它页面
F_RESUACTRECT	52H	从其它页面中恢复活动区域的内容
F_CLRSCREEN	56H	清除整个操作页面的显示内容
F_FILLRECT	58H	填充矩形区域
F_FILLACTRECT	59H	填充整个活动区域(其实是填充整个可视活动区域)
F_FILLRECTOUT	5AH	填充矩形以外的区域
F_DISPBYTE	5DH	显示文本像素
F_DISPBMPA	5EH	显示通用显示信息缓冲区内的位图
F_DISPBMPB	5FH	显示预置的位图
F_DISPCHR	60H	显示字符
F_DISPSTRA	61H	显示通用显示信息缓冲区内的字符串
F_DISPSTRB	62H	显示预置的字符串
F_DISPTXTA	63H	显示通用显示信息缓冲区内的字符串, 具有自动换行功能
F_DISPTXTB	64H	显示预置的字符串, 具有自动换行功能
F_DRAWPIXEL	67H	显示图形像素
F_DRAWHLINE	68H	绘制水平线
F_DRAWVLINE	69H	绘制垂直线
F_DRAWRECT	6AH	绘制单边矩形框
F_DRAWRECTSHD	6BH	绘制单边矩形框的右下边线
F_DRAWSHDRECT	6CH	绘制带阴影的单边矩形框
F_DRAWLINE	6DH	绘制任意两点之间的直线段
F_SHOWHSCOLL	70H	显示水平滚动条
F_SHOWVSCOLL	71H	显示垂直滚动条
F_DISPFIX16	74H	无符号十六位定点数值十进制显示
F_DISPFIX16	75H	十六位定点数值十进制显示
F_VIEWSPAGE	80H	显示静态页面
F_VIEWAPAGE	81H	显示动态页面
F_VIEWPAGE	82H	显示页面
F_VIEWOBJECT	84H	显示对象
F_SETFOCUSED	88H	设置对象的焦点信息
F_GETFOCUSED	89H	获得对象的焦点信息
F_GETOBJITEMS	8CH	获得单选框, 复选框, 列表框的选项数目或数值编辑器的有效位数
F_SETOBJCUR	8DH	设置单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号

F_GETOBJCUR	8EH	获得单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_INCOBJCUR	8FH	循环递增单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_DECOBJCUR	90H	循环递减单选框, 复选框, 列表框的选中项序号或数值编辑器光标位序号
F_SETCHECK	92H	设置复选框所有选项状态
F_GETCHECK	93H	获得复选框所有选项状态
F_ALTCHKISTA	94H	切换复选框某项状态
F_GETCHKISTA	95H	获得复选框某项状态
F_ALTCHKESTA	96H	切换复选框选中项状态
F_GETCHKESTA	97H	获得复选框选中项状态
F_INCLSTIS_V	9AH	限制递增列表框选中项行号
F_DECLSTIS_V	9BH	限制递减列表框选中项行号
F_SETSCVALUE	9EH	设置滚动条当前值和最大值
F_GETSCVALUE	9FH	获得滚动条当前值和最大值
F_SETVALFMT	A2H	设置动态数值框的显示格式
F_SETAVALUE	A3H	设置动态数值框的显示数据
F_SETSTDELMT	A6H	设置标准十进制定点数值编辑器的数值极限, 并重新调整编辑数值
F_SETSTDEVAL	A7H	设置标准十进制定点数值编辑器的数值
F_GETSTDEVAL	A8H	获得标准十进制定点数值编辑器的数值和极限值
F_SETCURVALUE	A9H	设置标准数值编辑器光标处数值
F_INCCURVALUE	AAH	标准数值编辑器光标处数值增一并且向前进位
F_DECCURVALUE	ABH	标准数值编辑器光标处数值减一并且向前借位
F_ALTSTDVESIGN	ACH	标准编辑器数值符号取反
F_GETATXT_LEN	B0H	获得动态文本框的长度和最大长度
F_SETATXT_XY	B1H	设置动态文本框显示内容的坐标
F_GETATXT_XY	B2H	获得动态文本框显示内容的坐标
F_ATXTCLR	B3H	清动态文本框的字符串内容
F_ATXTAPPEND	B4H	在动态文本框的末尾添加文本
F_ATXTINSERT	B5H	在动态文本框中插入文本
F_ATXTDELETE	B6H	在动态文本框中删除字符串