

XFD/SFD用户函数和指令集V05

- XFD模块内置python脚本,可支持脚本编程, SFD只支持插入用户函数, 不支持python编程
- XFD支持python的基本逻辑操作, SFD不支持
- SFD由于不支持脚本编程, 无法使用user.get_xx对应的用户函数
- 由于性能的原因, SFD不支持set_anim_xxx对应的用户函数
- XFD/SFD的用户函数列表

函数名称	函数内容	函数描述	示例代码	示例说明
重启指令	user.restart()	此指令用于重启模块, 如主机遇到某种意外情况重启时, 能通过复位指令让XFD和主机同步	<code>user.restart()</code>	重启模块
设置波特率	user.set_bps(rate)	设置模块的通信波特率 Rate对应波特率为: 115200/38400/19200/9600	<code>user.set_bps(9600)</code>	设置模块的波特率为9600
获取波特率	user.get_bps()	返回值分别是: 115200/38400/19200/9600	<code>rate=user.get_bps()</code>	将获取到的波特率的值保存到rate临时变量中
设置背光亮度	user.set_bl(n)	n是背光的亮度其范围为0~255, 0为最暗	<code>user.set_bl(100)</code>	设置背光的亮度为100的数值
获取背光亮度	user.get_bl()	返回值为背光的亮度值	<code>val=user.get_bl()</code>	将获取到的背光的亮度值保存到val临时变量中
设置x坐标	user.set_x(id,val)	id为当前页面的控件id, val为控件的x坐标	<code>user.set_x(1,20)</code>	设置控件1的x坐标为20
获取x坐标	user.get_x(id)	指定要获取的控件id, 返回值为x的坐标值	<code>val=user.get_x(1)</code>	将获取到的控件1的x坐标保存到val临时变量中
设置y坐标	user.set_y(id,val)	id为当前页面的控件的id, val为控件的y坐标	<code>user.set_y(1,20)</code>	设置控件1的y坐标为20
获取y坐标	user.get_y(id)	指定要获取的控件id, 返回值为x的坐标值	<code>val=user.get_y(1)</code>	将获取到的控件1的y坐标保存到val中
设置控件W宽度	user.set_width(id,val)	id为当前页面的控件id, val为控件的宽度	<code>User.set_width(1,20)</code>	将控件1的宽度设置为20
获取W宽度	user.get_width(id)	指定要获取的控件id, 返回值为宽度值	<code>val=user.get_width(1)</code>	将控件1的宽度保存到val中
设置H高度	user.set_height(id,val)	id为当前页面的控件id, val为控件的高度	<code>user.set_height(1,20)</code>	将控件1的高度设置为20
获取控件H高度	val=user.get_height(id)	指定要获取的控件id, 返回值为高度值	<code>val=user.get_height(1)</code>	将控件1的高度保存到val中
设置文本值	user.set_text(id,*str)	指定控件id和需要修改的文本内容str, 一般指定为label/textbox/qrcode这几个控件	<code>user.set_text(1,"正常")</code>	将控件1的文本修改为“正常”这两个字
获取控件文本	user.get_text(id)	指定控件id获取文本内容, 一般指定为label/textbox/qrcode这几个控件	<code>text=user.get_text(1)</code>	将控件1的文本的内容保存到text指针中
设置控件值	user.set_val(id,val)	指定id, 修改控件的值, 一般指定为用于prog/slider/arc/img这几个控件	<code>user.set_val(3,100)</code>	将控件3的值修改为100
获取控件的值	user.get_val(id)	指定控件id, 获取控件的值	<code>val=user.get_val(1)</code>	将获取到的控件1的值保存到val
设置蜂鸣器值	user.set_buzzer(time)	指定蜂鸣器的时间time(单位为毫秒), 当值为0时, 禁用蜂鸣器	<code>user.set_buzzer(100)</code>	设置蜂鸣器的蜂鸣时间为100毫秒
获取蜂鸣器的值	user.get_buzzer()	读取蜂鸣器的蜂鸣时间参数	<code>val=user.get_buzzer()</code>	将蜂鸣器的蜂鸣时间读取到val临时变量中
设置定时器	user.set_timer(id, one_time, time, enable)	XFD/SFD有8个通用定时器, 定时器id从0~7, 开启定时器需指定: 定时器id, 定时器类型选项、定时时间和使能开关	<code>user.set_timer(0,1,100,1)</code>	使能通定时器0为1次触发类型, 其触发时间为100ms
设置ivar	user.set_ivar(id,val)	指定id设置XFD内部的整型变量的值为val (id为0~31)	<code>user.set_ivar(0,20)</code>	设定IVAR[0]的值为20
获取ivar	user.get_ivar(id)	指定id获取IVAR[id]的值	<code>val=user.get_ivar(1)</code>	获取IVAR[1]的值保存到val临时变量中
设置svar	user.set_svar(id,str)	指定id设置的文本变量的值为str	<code>user.set_svar(0,"广东深圳")</code>	设SVAR[0]的内容为:“广东深圳”
获取svar	user.get_svar(id)	指定id获取SVAR[id]的文本内容	<code>str=user.get_svar(9)</code>	获取SVAR[9]的内容保存到str指针中
跳转页面	user.set_page(id,delay)	跳转到id的画面, 其中delay为等待delay毫秒后再跳转	<code>user.set_page(1,10)</code>	延时10ms后跳转到页面1
获取当前页面	user.get_page()	获取当前页面的id值	<code>id=user.get_page()</code>	获取当前页面的值并保存到id临时变量中
设置定时器的计数值	set_timer_times(id,cnt)	指定定时器的id, 可通过此指令设置定时器被调用的次数cnt	<code>user.set_timer_times(1,10)</code>	设定定时器1的计数值为10
获取定时器的计数值	get_timer_times(id)	指定定时器的id, 可通过此指令设置定时器被调用的次数	<code>cnt=user.get_timer_times(1)</code>	获取定时器1的计数值保存到cnt临时变量中

函数名称	函数内容	函数描述	示例代码	示例说明
		cnt		
使能485的地址	addr_enable(en)	设定XFD/SFD 485的地址模式	user.addr_enable(1)	使能XFD/SFD 485 串口的地址模式
设置485的地址	set_addr(addr)	设定XFD/SFD 485串口的地址	user.set_addr(1)	设置XFD/SFD 485串口的地址为1
使能数据传输的CRC16校验	set_crc16(en)	设定XFD/SFD 485的CRC16校验功能	user.set_crc16 (1)	设置XFD/SFD 485 串口的CRC16校验
隐藏控件	set_hidden(id,all,enable)	指定控件id， all 为全部隐藏， enable为使能开关	user.set_hidden (1,0,1)	隐藏控件1
设置背景色1	set_bg_color(id,state,color)	指定控件id， state为控件的状态， color为颜色值	user.set_bg_color(1,0,255)	设置控件1状态0下的颜色为蓝色
获取背景色1	get_bg_color(id,state)	指定控件id和state， 返回为背景色1的值	color=user.get_bg_color(1,0)	获取控件1状态0下的背景色1
设置背景色2	set_bg_color2(id,state,color)	指定控件id， state为控件的状态， color为颜色值	user.set_bg_color2(1,0,255)	设置控件1状态0下的颜色为蓝色
设置背景色2	get_bg_color2(id,state)	指定控件id和state， 返回为背景色1的值	color=user.get_bg_color2(1,0)	获取控件1状态0下的背景色1
设置字体景色	set_font_color(id,color)	指定控件id， color为颜色值	user.set_font_color(1,255)	设置控件1的颜色为蓝色
获取字体颜色	get_font_color(id)	指定控件id， 返回为字体景色的值	color=user.get_font_color(2)	获取控件2的字体颜色
设置透过率	set_opa(id,state,val)	指定控件id、状态和设定值， 设置控件的透明度	user.set_opa(2,0,0)	设置控件2在0状态下为全透明
获取透过率	get_opa(id,state)	指定控件id和state， 返回控件的透明度	val=user.get_opa(2,0)	获取控件2在状态0下的透明度保存到临时变量val中
设置空等时间	set_delay(ms)	指定等待时间ms， 用于空等	user.set_delay(100)	设置等待100ms
设置meter的值	set_mt_val(wid,val,state)	设置Meter控件wid的指针的编号state 为val的内容	user.set_mt_val(1,100,0)	将meter控件中的指针0的值设置为100
获取meter的值	get_mt_val(wid,state)	获取Meter控件wid的指针编号state的值	val=user.get_mt_val(1,0)	将meter控件1中的指针0的值读取到临时变量val中
保存gui的内容	save_gui(en)	当en=1时， 保存页面的修改	user.save_gui(1)	
隐藏曲线	chart_hidden(wid,ch,enable)	隐藏wid的chart控件中的ch通道， enable为使能开关	user.chart_hidden(1,1,1)	隐藏chart控件1中的1通道
设置休眠	set_sleep(time,bl_val)	指定休眠时间（单位为秒）和休眠后的背光亮度,如果time值为0， 即不进入休眠模式	user.set_sleep(5,10)	启动休眠时间为5秒， 休眠后的背光亮度为10
获取休眠时间	get_sleep_time()	获取XFD的休眠时间	val=user.get_sleep_time()	将休眠时间保存到临时变量val中
获取休眠时的背光亮度	get_sleep_bl()	获取XFD休眠时的背光亮度值	user.set_sleep_bl()	将休眠时指定的背光亮度保存到临时变量val中
设置动画	set_gif(wid,start_id,end_id,delay,loop_times,infinite,page,next_page,en)	Wid 为执行动画的IMG控件id start_id为图片的开始id end_id 为图的结束id delay 为执行动画的时延， 单位为毫秒 loop_times 指定动画的次数 infinite 是否为无限循环 page 指定动画执行所在的页面 next_page 表示执行完动画后跳转到对应的页面 en为动画使能开关	user.set_gif (2,0,10,100,10,0,0,3,1)	启动 page0控件2的动画功能， 指定的图片从0到10共11张， 动画的延时为100ms， 执行动画10次就跳转到页面3
控件自动上报坐标开关	report_pos(en)	按钮控件、panel控件在被触摸的时候能够上报位置坐标信息， 此功能可以通过设置en=1来实现， 如果想关闭自动上报坐标， en=0即可	user.report_pos(1)	使能控件的上报坐标功能
手动上报数据	report_data(header,index,data)	此用户函数用于通过串口发送数据给上位机， 其中header为数据头， 可以是0XAA,0xBB等任意的数据， index为索引值， 可以是0~255的任意值， data为32位整型数据， 为四个字节的任意数据	user.report_data(0xAA,1,0x01020304)	发送： 0XAA+1+01+02+03+04+FF FC FF FF 给上位机， 注意如果带地址和CRC的话， 数据会加入相应的地址和校验的内容一起发送
获取固件版本号	get_ver()	此指令没有参数， 返回字符串“XXX_YY”XXX为版本的内容， YY 为版本的编号 如获取到的内容为：“FWxx_SHMI_ST_01” 版本的内容为“FWxx_SHMI_ST” 版本号“01”	ver=user.get_ver()	获取当前的版本信息到临时变量ver中
设置隐藏当前页面	set_hidden_current_page(en)	此功能可以通过设置en=1来实现关闭当前页面显示的功能	user.set_hidden_current_page(1)	使能隐藏当前的页面
设置横向动画	set_anim_hdir(id,start,end,time,mode)	id为对应的控件id， start为x的起始坐标， end为x的结束坐标， time为动画的持续时间， 单位为毫秒， mode总共有7种： 0： 从头到尾的速度一样 1： 以低速开始 2： 以低速结束 3： 以低速开	user.set_anim_hdir(1,10,100,100,5)	控件1实现从10到100的横向移动， 移动的时间为100ms， 移动的方式为到终点后反弹

函数名称	函数内容	函数描述	示例代码	示例说明
		始低速结束 4：超过终点值 5：终点反弹一些 6 最后一步改变		
设置竖向动画	set_anim_vdir(id,start,end,time,mode)	id为对应的控件id, start为y的起始坐标, end为y的结束坐标, time为动画的持续时间, 单位为毫秒, mode总共有7种：0：从头到尾的速度一样 1：以低速开始 2：以低速结束 3：以低速开始低速结束 4：超过终点值 5：终点反弹一些 6 最后一步改变	user.set_anim_vdir(1,10,100,100,5)	控件1实现从10到100的纵向移动, 移动的时间为100ms, 移动的方式为到终点后反弹
设置任意方向的动画	set_anim_any(id,startx,endx,starty,endy,time,mode)	id为对应的控件id, startx为x的起始坐标, endx为x的结束坐标, starty为y的起始坐标, endy为y的结束坐标, time为动画的持续时间, 单位为毫秒, mode总共有7种：0：从头到尾的速度一样 1：以低速开始 2：以低速结束 3：以低速开始低速结束 4：超过终点值 5：终点反弹一些 6 最后一步改变	user.set_anim_any(4,-10,33,400,210,400,5)	控件4实现x从-10到33, y从400到210的左下角出现的移动, 时间为100ms, 移动的方式为到终点后反弹
设置曲线的值	set_ct_val(wid,ch,id,val,en)	Wid用于指定曲线的控件编号, ch指哪一条曲线(编号从0开始), id为曲线中某个点, val为要更新的某个点的值, 当en=1时, 用于使能更新	user.set_ct_val(1,1,20,1)	将控件1的的1号曲线中的点1的值设置为20
获取曲线的值	get_ct_val(wid,ch,id)	Wid用于指定曲线的控件编号, ch指哪一条曲线(编号从0开始), id指定要获取的曲线中的某个点	val=get_ct_val(1,0,1)	获取曲线控件1中的0号曲线中的点1的值, 存到临时变量val中
设置用户IO口的值	set_usr_pin(id,val)	XFD提供5个用户使用的GPIO口, 通过set_usr_pin能够设置对应的IO口(目前设置的是5个口, id从0~4) 为1或者0	user.set_usr_pin(0,1)	设置用户gpio口0的值为1
获取用户IO口的值	get_usr_pin(id)	XFD提供5个用户使用的GPIO口, 通过get_usr_pin能够获取对应的IO口(目前设置的是5个口, id从0~4) 的值	val=user.get_usr_pin(0)	获取用户gpio口0的值保存到临时变量val中
串口发送字符串str	report_str(str)	将字符串str通过串口通讯口发送出去	From gb2312 import Gb2312 gcode = Gb2312() r = gcode.strs('我爱深圳!') uesr.report_str(r)	将UTF-8的'我爱深圳!'转成Gb2312编码后通过串口发送出去
控件值自增	inc_val(id,val)	将控件id的值自增val的步长	user.inc_val(1,1)	将控件的值在原来的基础上自增1
控件值自减	dec_val(id,val)	将控件id的值自减val的步长	user.dec_val(1,1)	将控件的值在原来的基础上自减1
Meter值自增	inc_mt_val(id,val,state)	将编号为id的meter中对应state的指针值自增val的值	user.inc_mt_val(1,1,0)	将id为1的meter控件中的0号指针值自增1
Meter值自减	dec_mt_val(id,val,state)	将编号为id的meter中对应state的指针值自减val的值	user.dec_mt_val(1,1,0)	将id为1的meter控件中的0号指针值自减1
背光自增	inc_bl(val)	将背光的值自增val的值	user.inc_bl(1)	将背光的值自增1
背光自减	dec_bl(val)	将背光的值自减val的值	user.dec_bl(1)	将背光的值自减1
标签值自增	inc_label_val(id,val,limit)	将标签中设定的数字类型的字符串的值自增val的值, 其中limit是自增的上限值	user.inc_label_val(1,1,100)	将标签控件1中的数字类型的字符串的值自增1, 当自增到100时, 不再增加
标签值自减	dec_label_val(id,val,limit)	将标签中设定的数字类型的字符串的值自减少val的值, 其中limit是自减的下限值	user.dec_label_val(1,1,0)	将标签控件1中的数字类型的字符串的值自减1, 当自增减到0时, 不再减少
控件触摸使能	touch_enable(id,en)	当en为1时, 使能编号id的控件的触摸功能	touch_enable(1,1)	使能控件1的触摸功能
设置菜单光标信息	set_cursor_info(width,color,opa,num,"map")	当不带触摸屏时, 通过这个用户函数能够设置界面中高亮显示的控件数量、光标的宽度、颜色、不透明度等信息, 一般在页面进入事件中加载此信息	user.set_cursor_info(2,16777215,255,2,"12/13")	将页面中的控件12和13, 设置为光标选中的范围, 当12或者13被选中时, 光标的宽度为2, 颜色为白色, 不透明度为255
切换页面加载方向函数	swi_page_load_dir(dir)	用于动态修改页面加载的方向, dir共有8个方向0~7, 具体参考SHMI加载方式中的方向选项	user.swi_page_load_dir(0)	将页面加载的方向修改成从左到右
自增菜单光标的值	set_cursor_inc()	当无触摸屏时, 通过设置菜单光标信息后, 此函数用于改变菜单光标(外边框高亮显示的控件)的值	user.set_cursor_inc()	增加当前页面中高亮显示的控件id序号
自减菜单光标的值	set_cursor_dec()	当无触摸屏时, 通过设置菜单光标信息后, 此函数用于改变菜单光标(外边框高亮显示的控件)的值	user.set_cursor_dec()	减少当前页面中高亮显示的控件id序号
获取菜单光标的值	get_cursor()	此函数用于获取当前的菜单光标的值	Val=user.get_cursor()	获取当前的菜单光标的值保存到val临时变量中
设置曲线的下一个值	set_ct_val2(wid,cid,val)	此函数用于设置曲线的下一个值, 通过设置这个值能够	user.set_ct_val2(1,0,20)	设置曲线控件1的0号曲线的下一个值为20

函数名称	函数内容	函数描述	示例代码	示例说明
		实现循环滚动曲线的效果		

- python的语法 (详细的内容参考：<https://www.runoob.com/python/python-basic-syntax.html>)

条目	说明	示例代码	示例说明
基础语法1	学习 Python 与其他语言最大的区别就是, Python 的代码块不使用大括号 {} 来控制类, 函数以及其他逻辑判断。python 最具特色的就是用缩进来写模块。 缩进的空白数量是可变的, 但是所有代码块语句必须包含相同的缩进空白数量, 这个必须严格执行。	if True: print ("True") else: print ("False")	
基础语法2	python 的注释使用#, 而不使用//	#!/usr/bin/python # -*- coding: UTF-8 -*- # 文件名: test.py if True: print ("Answer") print ("True") else: print ("Answer") # 没有严格缩进, 在执行时会报错 print ("False")	其中下面的内容均为注释部分, 不参与运行: #!/usr/bin/python # -*- coding: UTF-8 -*- # 文件名: test.py # 没有严格缩进 # 没有严格缩进
运算	python 的变量不需要申明 能直接参与运算	a=10 b=20 c="串口" d="显示模块" print(a+b) print(c+d)	运行python后, 串口输出 30 和"串口显示模块"
条件语句	if 判断条件: 执行语句..... else: 执行语句.....	num = 10 if num < 0 or num > 10: print ('hello') else: print ('undefine')	运行python后, 串口输出 undefine
while循环语句	while 判断条件(condition): 执行语句(statements).....	count = 0 while (count < 9): print 'The count is:', count count = count + 1 print "Good bye!"	输出结果: The count is: 0 The count is: 1 The count is: 2 The count is: 3 The count is: 4 The count is: 5 The count is: 6 The count is: 7 The count is: 8 Good bye!

XFD/SFD指令集

- 指令集必须通过串口以16进制数据流的方式发送给模块
- 系统控制指令

指令名称	描述	指令码1	指令码2	参数1	参数2	参数3	参数4	例子1 (以地址为0002, 带CRC16, 参数1的值为1如参数2有效时, 其值及参数2以上的值都为2为例子)	例子2 (以无地址, 不带CRC16, 参数1的值为1, 参数2及以上的值都为2为例子)
restart_device	复位系统	0x03	-	42330	21301	-	-	EE { 00 02} 03 A5 5A 53 35 {38 9E} FF FC FF FF	EE 00 02 03 A5 5A 53 35 FF FC FF FF
shake_hand	握手	0x04	-	-	-	-	-	EE { 00 02} 04 {83 BE} FF FC FF FF	EE 04 FF FC FF FF
set_page	跳转到特定的页面	0x05	-	pid	times(ms)	-	-	EE { 00 02} 05 00 01 00 02 {01 38} FF FC FF FF	EE 05 00 01 00 02 FF FC FF FF
get_page	获取当前页面	0x06	-	-	-	-	-	EE { 00 02} 06 {42 3F} FF FC FF FF	EE 06 FF FC FF FF
set_bl	设置背光亮度	0x07	-	bl	-	-	-	EE { 00 02} 07 00 01 {C1 01} FF FC FF FF	EE 07 00 01 FF FC FF FF
get_bl	获取背光亮度	0x08	-	-	-	-	-	EE { 00 02} 08 {86 BE} FF FC FF FF	EE 08 FF FC FF FF
set_bps	设置波特率	0x09	-	bps	-	-	-	EE { 00 02} 09 00 01 {02 60} FF FC FF FF	EE 09 00 01 FF FC FF FF
get_bps	获取波特率	0x0a	-	-	-	-	-	EE { 00 02} 0A {47 3F} FF FC FF FF	EE 0A FF FC FF FF
set_timer	开关timer	0x0b	-	tid	one_shot	time(ms)	enable	EE { 00 02} 0B 00 01 00 02 00 02 00 02 {C1 30} FF FC FF FF	EE 0B 00 01 00 02 00 02 00 02 FF FC FF FF
save_gui	保存控件的内容	0x0c	-	-	-	-	-	EE { 00 02} 0C {45 BF} FF FC FF FF	EE 0C FF FC FF FF
save_sys	保存系统参数	0x0d	-	-	-	-	-	EE { 00 02} 0D {85 7E} FF FC FF FF	EE 0D FF FC FF FF
save_xvar	保存XVAR变量	0x0e	-	-	-	-	-	EE { 00 02} 0E {84 3E} FF FC FF FF	EE 0E FF FC FF FF
set_delay	设置延时时间	0x0f	-	time(ms)	-	-	-	EE { 00 02} 0F 00 01 {03 80} FF FC FF FF	EE 0F 00 01 FF FC FF FF
set_sleep	设置休眠	0x10	-	time(s)	bl_val	-	-	EE { 00 02} 10 00 01 00 02 {C2 35} FF FC FF FF	EE 10 00 01 00 02 FF FC FF FF
get_sleep	获取休眠参数	0x11	-	-	-	-	-	EE { 00 02} 11 {4C 7F} FF FC FF FF	EE 11 FF FC FF FF
set_buzzer	设置buzzer参数	0x12	-	time(s)	-	-	-	EE { 00 02} 12 00 01 {05 10} FF FC FF FF	EE 12 00 01 FF FC FF FF

指令名称	描述	指令码1	指令码2	参数1	参数2	参数3	参数4	例子1 (以地址为0002, 带CRC16, 参数1的值为1如参数2有效时, 其值及参数2以上的值都为2为例子)	例子2 (以无地址, 不带CRC16, 参数1的值为1, 参数2及以上的值都为2为例子)
get_buzzer	获取buzzer参数	0x13	-	-	-	-	-	EE { 00 02} 13 {8D FE} FF FC FF FF	EE 13 FF FC FF FF
set_ivar	设置ivar的值	0x14	-	xid	valH	valL	-	EE { 00 02} 14 00 01 00 02 00 02 {C0 52} FF FC FF FF	EE 14 00 01 00 02 00 02 FF FC FF FF
get_ivar	获取ivar的值	0x15	-	xid	-	-	-	EE { 00 02} 15 00 01 {C4 A1} FF FC FF FF	EE 15 00 01 FF FC FF FF
set_svar	设置svar的值	0x16	-	xid	str	-	-	EE { 00 02} 16 00 01 32 00 {63 29} FF FC FF FF	EE 16 00 01 32 00 FF FC FF FF
get_svar	获取svar的值	0x17	-	xid	-	-	-	EE { 00 02} 17 00 01 {04 00} FF FC FF FF	EE 17 00 01 FF FC FF FF
get_ver	获取版本号	0x18	-	-	-	-	-	EE { 00 02} 18 {4A BF} FF FC FF FF	EE 18 FF FC FF FF
set_cali	校正电阻触摸屏	0x1B	-	-	-	-	-	EE { 00 02} 1B {4B FF} FF FC FF FF	EE 1B FF FC FF FF

• UI相关设置指令

指令名称	描述	指令码1	指令码2	参数1	参数2	参数3	参数4	例子1 (以地址为0002, 带CRC16, 参数1的值为1如参数2有效时, 其值及参数2以上的值都为2为例子)	例子2 (以无地址, 不带CRC16, 参数1的值为1, 参数2及以上的值都为2为例子)
set_obj_x	设置控件的X坐标	0x81	0x01	wid	x_val	-	-	EE { 00 02} 81 01 00 01 00 02 {CB F3} FF FC FF FF	EE 81 01 00 01 00 02 FF FC FF FF
set_obj_y	设置控件的Y坐标	0x81	0x02	wid	y_val	-	-	EE { 00 02} 81 02 00 01 00 02 {CB B7} FF FC FF FF	EE 81 02 00 01 00 02 FF FC FF FF
set_obj_width	设置控件的W宽度	0x81	0x03	wid	width	-	-	EE { 00 02} 81 03 00 01 00 02 {0B 8A} FF FC FF FF	EE 81 03 00 01 00 02 FF FC FF FF
set_obj_height	设置控件的H高度	0x81	0x04	wid	height	-	-	EE { 00 02} 81 04 00 01 00 02 {CB 3F} FF FC FF FF	EE 81 04 00 01 00 02 FF FC FF FF
set_obj_text	设置控件文本内容	0x81	0x05	wid	str	-	-	EE { 00 02} 81 05 00 01 32 00 {AA 96} FF FC FF FF	EE 81 05 00 01 32 00 FF FC FF FF
set_obj_bgcolor1	设置控件的背景色1	0x81	0x06	wid	state	bcolor1H	bcolor1L	EE { 00 02} 81 06 00 01 00 02 00 02 {8B 44} FF FC FF FF	EE 81 06 00 01 00 02 00 02 FF FC FF FF
set_obj_bgcolor2	设置控件的背景色2	0x81	0x07	wid	state	bcolor2H	bcolor2L	EE { 00 02} 81 07 00 01 00 02 00 02 {1B 49} FF FC FF FF	EE 81 07 00 01 00 02 00 02 FF FC FF FF
set_obj_font_color	设置控件的前景色	0x81	0x08	wid	fcolorH	fcolorL	-	EE { 00 02} 81 08 00 01 00 02 00 02 {A6 1D} FF FC FF FF	EE 81 08 00 01 00 02 00 02 FF FC FF FF
set_obj_val	设置控件的值	0x81	0x09	wid	val	-	-	EE { 00 02} 81 09 00 01 00 02 {0A 12} FF FC FF FF	EE 81 09 00 01 00 02 FF FC FF FF
set_obj_hidden	设置隐藏控件	0x81	0x0a	wid	all	enable	-	EE { 00 02} 81 0A 00 01 00 02 00 02 {66 3E} FF FC FF FF	EE 81 0A 00 01 00 02 00 02 FF FC FF FF
set_chart_hidden	设置隐藏曲线	0x81	0x0b	wid	ch	enable	-	EE { 00 02} 81 0B 00 01 00 02 00 02 {A6 2E} FF FC FF FF	EE 81 0B 00 01 00 02 00 02 FF FC FF FF
set_chart_val	设置曲线的值	0x81	0x0c	wid	ch	pid	num	EE { 00 02} 81 0C 00 01 00 02 00 02 00 02 00 02 {D1 5C} FF FC FF FF	EE 81 0C 00 01 00 02 00 02 00 02 00 02 FF FC FF FF
set_hidden_current_page	设置隐藏当前页面	0x81	0x0d	en	-	-	-	EE { 00 02} 81 0D 00 01 {0B 78} FF FC FF FF	EE 81 0D 00 01 FF FC FF FF
set_meter_val	设置meter的指针值	0x81	0x0e	wid	val	state	-	EE { 00 02} 81 0E 00 01 00 02 00 02 {A6 7B} FF FC FF FF	EE 81 0E 00 01 00 02 00 02 FF FC FF FF
set_touch_enable	设置控件触摸使能	0x81	0x0f	wid	en	-	-	EE { 00 02} 81 0F 00 01 00 01 {0B DA} FF FC FF FF	EE 81 0F 00 01 01 01 FF FC FF FF
set_widget_status	设置触摸控件的状态	0x81	0x10	wid	status	-	-	EE { 00 02} 81 10 00 01 00 01 {C9 4F} FF FC FF FF	EE 81 10 00 01 01 01 FF FC FF FF
set_cursor_inc	自增菜单光标值	0x81	0x11	-	-	-	-	EE { 00 02} 81 11 {EC A1} FF FC FF FF	EE 81 11 FF FC FF FF

指令名称	描述	指令码1	指令码2	参数1	参数2	参数3	参数4	例子1 (以地址为0002, 带CRC16, 参数1的值为1如参数2有效时, 其值及参数2以上的值都为2 为例子)	例子2 (以无地址, 不带CRC16, 参数1的值为1, 参数2及以上的值都为2 为例子)
set_cursor_dec	自减菜单光标值	0x81	0x12	-	-	-	-	EE {00 02} 81 12 {ED E1} FF FC FF FF	EE 81 12 FF FC FF FF
set_ct_val2	设置曲线的下一个值	0x81	0x13	wid	cid	val	-	EE {00 02} 81 13 00 01 00 00 00 02 {67 17} FF FC FF FF	EE 81 13 00 01 00 00 00 02 FF FC FF FF

• UI相关读取指令

指令名称	描述	指令码1	指令码2	参数1	参数2	参数3	参数4	例子1 (以地址为0002, 带CRC16, 参数1的值为1如参数2有效时, 其值及参数2以上的值都为2 为例子)	例子2 (以无地址, 不带CRC16, 参数1的值为1, 参数2及以上的值都为2 为例子)
get_obj_x	获取控件的X坐标	0x82	0x01	wid	-	-	-	EE {00 02} 82 01 00 01 {5C B8} FF FC FF FF	EE 82 01 00 01 FF FC FF FF
get_obj_y	获取控件的Y坐标	0x82	0x02	wid	-	-	-	EE {00 02} 82 02 00 01 {5C 48} FF FC FF FF	EE 82 02 00 01 FF FC FF FF
get_obj_width	获取控件的宽度	0x82	0x03	wid	-	-	-	EE {00 02} 82 03 00 01 {9C 19} FF FC FF FF	EE 82 03 00 01 FF FC FF FF
get_obj_height	获取控件的高度	0x82	0x04	wid	-	-	-	EE {00 02} 82 04 00 01 {5D A8} FF FC FF FF	EE 82 04 00 01 FF FC FF FF
get_obj_text	获取控件的文本	0x82	0x05	wid	-	-	-	EE {00 02} 82 05 00 01 {9D F9} FF FC FF FF	EE 82 05 00 01 FF FC FF FF
get_obj_bgcolor1	获取控件的bgcolor1	0x82	0x06	wid	state	-	-	EE {00 02} 82 06 00 01 00 02 {38 46} FF FC FF FF	EE 82 06 00 01 00 02 FF FC FF FF
get_obj_bgcolor2	获取控件的bgcolor2	0x82	0x07	wid	state	-	-	EE {00 02} 82 07 00 01 00 02 {F8 7B} FF FC FF FF	EE 82 07 00 01 00 02 FF FC FF FF
get_obj_font_color	获取控件的font_color	0x82	0x08	wid	-	-	-	EE {00 02} 82 08 00 01 {5E 68} FF FC FF FF	EE 82 08 00 01 FF FC FF FF
get_obj_val	获取控件的值	0x82	0x09	wid	-	-	-	EE {00 02} 82 09 00 01 {9E 39} FF FC FF FF	EE 82 09 00 01 FF FC FF FF
get_chart_val	获取曲线的值	0x82	0x0a	wid	sid	pid	num	EE {00 02} 82 0A 00 01 00 02 00 02 00 02 {84 E1} FF FC FF FF	EE 82 0A 00 01 00 02 00 02 00 02 FF FC FF FF
get_meter_val	获取meter的指针值	0x82	0x0b	wid	state	-	-	EE {00 02} 82 0B 00 01 00 02 {F9 6B} FF FC FF FF	EE 82 0B 00 01 00 02 FF FC FF FF
get_touch_enable	获取控件触摸使能状态	0x82	0x0c	wid	-	-	-	EE {00 02} 82 0C 00 01 {9F 29} FF FC FF FF	EE 82 0C 00 01 FF FC FF FF
get_widget_status	获取触摸控件的状态	0x82	0x0D	wid	-	-	-	EE {00 02} 82 0D 00 01 {5F 78} FF FC FF FF	EE 82 0D 00 01 FF FC FF FF
get_cursor	获取当前菜单光标值	0x82	0x0E	-	-	-	-	EE {00 02} 82 0E 00 01 {D4 E0} FF FC FF FF	EE 82 0E FF FC FF FF