串口 HMI 指令集

发布版本 V2.6 版本修改日期:2017 年 11 月 13 日

注：

1. 设备接受指令结束符为"0XFF 0XFF 0XFF"三个字节。
2. 所有指令名以及参数全部使用 ASCII 字符串格式，非二进制数据，便于阅读和调试。
3. 所有指令名使用小写字母(此处仅仅指的是指令名称为小写，参数该大写的时候还是要大写)。
4. **0.39 版本开始，在运行中修改控件的任何属性都将自动刷新，不再需要使用手动刷新指令。(原来的版本在修改非加粗属性时需要手动刷新)。**

分类一:对象及系统操作指令

1.page 刷新页面
page pageid pageid:页面 ID 或页面名称 实例 1:page 0 (刷新 ID 为 0 的页面) 实例 2:page main (刷新名称为 main 的页面) 备注： 1.设备上电自动刷新第 0 页。 2.也可以对系统变量 dp 赋值来实现跳转页面(如 dp=0)，系统变量 dp 可以设置可以读取，具体请参看系统变量列表。
2.ref 重绘控件
ref obj obj:控件 ID 或控件名称 实例 1:ref 1 (重绘 ID 为 1 的控件) 实例 2:ref t0 (重绘名称为 t0 的控件) 备注： 如果一个控件被 GUI 指令画出来的内容遮挡或者被另外的控件遮挡之后需要再显示出来，就使用 ref 来重绘。
3.click 激活控件的按下/弹起事件
click obj,event obj:控件 ID 或控件名称 event:事件序号:0 为弹起，1 为按下 click b0,1 (激活名称为 b0 的控件的按下事件) click 2,0 (激活 ID 为 2 的控件的弹起事件) 备注： 控件的按下/弹起事件在屏幕上触摸的时候会自动激活，如果在没有触摸的情况下想要手动激活，就使用 click 指令即可。
4.get 带格式获取变量值/常量值
get att att:变量名称 实例 1:get t0.txt (返回控件 t0 的 txt 属性值) 实例 2:get j0.val(返回控件 j0 的 val 属性值)

深圳市淘晶驰电子有限公司 http://tjc1688.com/
地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

实例 3:get “123”(返回常量字符串”123”) 实例 4:get 123(返回常量数值:”123”)

备注:

1.使用 get 指令获取的变量为字符串类型时，返回的数据为 0X70+字符串内码+结束符，如果是数值类型(如进度条的 val 属性)设备返回 0X71+变量的 4 字节十六进制数据(int 类型)+结束符。数值的存放模式为小端模式（即低位在前，高位在后)。

2.get 指令可以由串口发送，也可以在上位软件编辑界写进用户代码中实现屏幕主动发送变量(主动发送的时候可以配合 printh 指令在前面加一段自定义标示来告诉单片机此变量是属于哪个控件的)。

3.get 指令和 print 指令很类似,唯一的区别是 get 返回的数据带了起始标示符（0x70 或 0x71）和结束符(0xff 0xff 0xff)，而 print 没有。

4..数据具体返回格式请查看本表格后面的”串口 HMI 设备返回数据格式”。

5.prints 将一个变量/常量从串口发送出去

prints att,lenth

att:变量名称

lenth:发送长度(0 为自动长度)

实例 1: prints t0.txt,0 (发送控件 t0 的 txt 属性值,长度为实际长度)

实例 2: prints j0.val,0(发送控件 j0 的 val 属性值,默认长度为 4 字节整形数据，小端模式储存)

实例 3: prints “123”,0(发送常量字符串”123”即：0x31 0x32 0x33)

实例 4: prints 123,0(发送常量数值：123 即:0x7b 0x00 0x00 0x00)

实例 5: prints 123,1(发送常量数值：123 的低 1 位数据 即:0x7b)

备注:

1.使用 prints 发送的变量为字符串类型时，设备直接返回字符串内码，如果是数值类型(如进度条的 val 属性)设备直接返回变量的 4 字节整形数据(Hex 数据，储存方式为小端模式，即低位在前)。

2.使用 prints 指令获取数据的时候，设备仅仅只发送数据内容，没有起始标示符，也没有结束符。

3.prints 指令可以配合 printh 指令在前面加一段自定义标示来告诉单片机此变量是属于哪个控件的)。

4. prints 指令和 get 指令很类似,区别是 get 发送的数据带了起始标示符（0x70 或 0x71）和结束符(0xff 0xff 0xff)，而 prints 没有,不过 prints 可以在后面继续用 printh 语句来加任何自定义标识符。

6.printh 让设备的串口发送自定义 16 进制 byte

printh hex

hex:需要发送的字符的 16 进制字符串表达式

实例:printh d0 a0(让设备发送 0xd0 0xa0 两个字节)

备注:

1.使用 printh 指令发送数据的时候，设备仅仅只发送指定的字符，不会发起始符，不会发空格,不会发结束符。

2.参数中每组字符间必须有且只能有一个空格隔开，16 进制的字符串表达式大小写均支持。

7.vis 隐藏/显示控件

vis obj,state

obj:控件名称或控件 ID

state:状态(0 或 1)

实例 1:vis b0,0 (隐藏 b0 控件) 实例 2:vis b0,1 (显示 b0 控件)

实例 3: vis 1,0 (隐藏 ID 为 1 的控件) 实例 4: vis 1,1 (显示 ID 为 1 的控件)

备注:

第一个参数 为 255 表示 当前页面所有控件，例:vis 255,0(隐藏当前页面所有控件) vis 255,1(显示当前页面所有控件)。

深圳市淘晶驰电子有限公司 http://tjc1688.com/

地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

8. tsw 控件触摸使能

tsw obj,state
obj:控件名称或控件 ID
state:状态(0 或 1)
实例 1:tsw b0,0 (让名称为 b0 的控件触摸失效) 实例 2:tsw b0,1 (让名称为 b0 的控件触摸有效)
实例 3:tsw 1,0 (让 ID 为 1 的控件触摸失效) 实例 4:tsw 1,1 (让 ID 为 1 的控件触摸有效)
备注:
第一个参数 为 255 表示 当前页面所有控件，例:tsw 255,0(当前页面所有控件触摸失效) tsw 255,1(当前页面所有控件触摸有效)。

9. randset 随机数范围设置

randset minval,maxval
minval:最小值
maxval:最大值
实例:ranset 1,100 (设置当前随机数产生范围为最小 1，最大 100)
备注:
1. 使用随机数之前需要先使用 randset 指令设定一次随机数产生范围，如果不设置，默认是最小 0，最大 2147483647。设置完范围以后，每读取一次系统变量 rand 将会得到一个随机数。
2. 使用 randset 指令每设定一次范围，将一直有效，直到重新上电或者设备复位才会恢复默认。
3. 随机数设定范围的数据类型为 int 类型(即：最小-2147483648,最大 2147483647)。

10. add 往曲线控件添加数据

add objid,ch,val
objid:曲线控件 ID 序号(此处必须是 ID 号，不支持使用控件名称)
ch:曲线控件通道号
val:数据 (最大 255，最小 0)
实例 1:add 1,0,30 (往 ID 为 1 的曲线控件的 0 通道添加数据 30)
实例 2:add 1,1,n0.val (往 ID 为 1 的曲线控件的 1 通道添加数据 n0.val)
备注:
1. 曲线数据只支持 8 位数据，最小 0，最大 255。
2. 每个 page 页面最多支持 4 个曲线控件,每个曲线控件最多支持 4 个通道。可以连续发送数据，控件会自动平推显示数据.在发送数据的过程中也可以随时修改控件属性，比如随时修改各个通道的前景色或背景色。

11. cle 清除曲线控件中的数据

cle objid,ch
objid:曲线控件 ID 序号(此处必须是 ID 号，不支持使用控件名称)
ch:曲线控件通道号(255 表示所有通道)
实例 1:cle 1,0 (清除 ID 为 1 的曲线控件的 0 通道数据)
实例 2:cle 1,255 (清除 ID 为 1 的曲线控件的所有通道数据)
备注:
1. 通道号为 255 时表示清除此曲线控件内的所有通道数据。

12. addt 曲线数据透传指令

addt objid,ch,qyt
objid: 曲线控件 ID 序号(此处必须是 ID 号，不支持使用控件名称)
ch:曲线控件中的通道号

qyt:本次透传数据的点数量

实例:addt 1,0,100 （ID 为 1 的曲线控件进入数据透传模式，透传点数为 100 点）

备注：

1. 曲线数据只支持 8 位数据，最小 0，最大 255。单次透传数据量最大 1024 字节

2. 发完透传指令后，用户需要等待设备响应才能开始透传数据，设备收到透传指令后，准备透传初始化数据大概需要 5ms 左右(如果在透传指令执行前串口缓冲区还有很多别的指令，那时间会更长)，设备透传初始化准备好以后会发送一个透传就绪的数据给用户（0XFE+结束符），表示设备已经准备好，此时可以开始发送透传数据。透传数据为纯 16 进制数据，不再使用字符串，也不再需要结束符,设备收完指定的数据量以后，才会恢复指令接收状态。否则一直处于数据透传状态，透传数据完成以后,设备会发送结束标记给用户（0XFD+结束符）。

3. 在指定的透传数量传输完成以前，曲线不会刷新，透传完毕之后会立即自动刷新。

13. doevents 转让系统控制权给屏幕刷新

doevents

实例：doevents （此指令不需要参数）

备注：

1. 在一个较多指令的过程执行中，或者在一个较长时间的循环语句中，系统所有控制权被此过程全部占用，在过程结束之前，尽管相应的内存数据可以任意正常读写，但是屏幕不会刷新显示，加入 doevents 后可以转让控制权给屏幕刷新，执行 doevents 之后，屏幕会刷新所有被改变过的控件，刷新完之后，控制权交回当前过程继续执行。防止屏幕呈现假死的显示状态。

2. doevents 多数情况下是配合 while 或 for 语句使用，使用方法请参看 while 或 for 语句的实例。

14. sendme 发送当前页面 ID 号到串口

sendme

实例 1：sendme （此指令不需要参数）

备注:

设备收到此指令会立刻把当前页面的 ID 号发送到串口，如果想要每次刷新页面自动发送页面 ID,请在页面的初始化事件里写上 sendme 语句即可。发送格式请参看本表格后面的”串口 HMI 设备返回数据格式”表格。

15. covx 变量类型转换

covx att1,att2,lenth,format

att1:源变量

att2:目标变量

lenth:字符串的长度(0 为自动长度，非 0 为固定长度)

format:申明数值类型(0-数字;1-货币;2-Hex)

实例 1:covx h0.val,t0.txt,0,0 （把滑块 h0 的 val 数值变量转换成 10 进制数字子符串并赋值给文本 t0 的 txt 变量,长度为自动)

实例 2:covx t0.txt,h0.val,0,0 (把文本 t0 的 txt 十进制数字字符串变量转换为数值并赋值给滑块 h0 的 val 数值变量,长度为自动)

备注:

1. lenth 始终表示的是字符串长度，数值转字符串的时候是目标变量的长度，字符串转数值的时候是源变量长度。

2. 如果目标变量和源变量类型相同，转换失败。

16. strlen 字符串变量字符长度测试

strlen att0,att1

att0:需要测试的字符串变量

att1:把测试结果赋值给此变量

实例:strlen t0.txt,n0.val (把字符串变量 t0.txt 的实际字符长度赋值给 n0.val)

备注:

1.strlen 测试的是以字符为单位的长度，而 btlen 测试的是以字节为单位的长度,比如一个汉字用 btlen 测试出来的长度是 2 字节,用 strlen 测试出来的长度是 1 字符。

2.被测试的变量必须是字符串类型，写入的变量必须是数值类型，否则会报错。

17.btlen 字符串变量字节长度测试

btlen att0,att1

att0:需要测试的字符串变量

att1:把测试结果赋值给此变量

实例:btlen t0.txt,n0.val (把字符串变量 t0.txt 的实际字节长度赋值给 n0.val)

备注:

1.btlen 测试的是以字节为单位的长度，而 strlen 测试的是以字符为单位的长度，比如一个汉字用 btlen 测试出来的长度是 2 字节,用 strlen 测试出来的长度是 1 字符。

2.被测试的变量必须是字符串类型，写入的变量必须是数值类型，否则会报错。

18.substr 字符串截取

substr att0,att1,star,lenth

att0:源变量(必须是字符串变量)

att1:目标变量(必须是字符串变量)

star:在源变量中的字符起始位置

lenth:截取字符串长度

实例:substr t0.txt,t1.txt,0,2 (从 t0.txt 中的 0 位置开始截取 2 个字符赋值给 t1.txt)

19.touch_j 触摸校准

touch_j

实例 1：touch_j (进入触摸校准功能,此指令不需要参数)

备注：

所有设备出厂时已经校准过，一般情况下不需要使用此功能。

20.ref_stop 暂停屏幕刷新

ref_stop

实例:ref_stop (此指令不需要参数)

备注：

1.暂停屏幕刷新之后，所有语句会继续解析并执行，相应的属性赋值操作也会正常运行，但是屏幕上的控件不会刷新，修改任何控件的任何属性都不会自动刷新显示（但是属性已经被正常修改了）。直到设备收到恢复刷新指令(ref_star)后，被修改过的控件将会立刻刷新显示。

2.暂停刷新之后，即便使用 ref 指令也不会立刻刷新，直到执行 ref_star 指令的时候统才会统一刷新，但是所有的 gui 绘图指令(比如画点，划线，画圆等)是不受影响的，会立即显示。

21.ref_star 恢复屏幕刷新

ref_star

实例:ref_star (此指令不需要参数)

备注：

此指令和 ref_stop 配合使用。

22.com_stop 暂停串口指令执行

com_stop

深圳市淘晶驰电子有限公司 http://tjc1688.com/

地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

实例 1:com_stop （此指令不需要参数）

备注：

1. 暂停串口指令执行之后设备会继续接受指令，但是都不会执行，全部放在指令缓存区，直到收到”com_star”指令后，设备会从暂停时的指令开始到当前为止的所有指令全部执行。

2. 使用指令暂停与恢复功能的时候，请评估您的设备的串口缓存区大小和指令缓存队列的最大数量是否足够支持你需要缓存的指令数目。这两项参数在你购买的设备规格书中的参数表中可以查询到。

23. com_star 恢复串口指令执行

com_star

实例 1:com_star （此指令不需要参数）

备注：

1. 设备收到此指令之后，将从暂停时的指令开始到当前为止的所有指令全部执行。

2. 使用指令暂停与恢复功能的时候，请评估您的设备的串口缓存区大小和指令缓存队列的最大数量是否足够支持你需要缓存的指令数目。这两项参数在你购买的设备规格书中的参数表中可以查询到。

24. code_c 清空串口指令缓冲区中还没有执行的所有指令

code_c

实例 1: code_c （此指令不需要参数）

立即清空串口指令缓冲区还没有执行的所有指令。

25. rest 复位

rest

实例:rest （此指令不需要参数）

26. wepo 写入一个变量到用户存储区(EEPROM)

仅增强型及以上系列适用

wepo att,add

att:变量/常量

add: 用户存储区位置(从 0 开始)

实例 1:wepo t0.txt,10 （将 t0.txt 的内容写入用户存储区的第 10 位置,在储存区中的占用空间为 t0.txt 的最大设置值+1，即 t0 的 txt-maxl 属性表示的大小+1)

实例 2:wepo “abcd”,10 （将字符串“abcd”写入用户存储区的第 10 位置，在储存区中占用大小为 5 字节)

实例 3:wepo 125,10 （将数值 125 写入用户存储区的第 10 位置, 在储存区中占用大小为 4 字节)

备注:

1. 写入内容为变量字符串的时候，在储存区中的占用空间为此变量的最大字符数+1；写入内容为常量字符串的时候，在储存区中的占用空间为此常量字符串的实际字符数+1。

2. 写入内容为变量数值或常量数值的时候，在储存区中的占用空间统一为 4 字节。

3. 使用用户存储区读写操作过程中请切记规划好数据区位置，以免位置交错引起数据覆盖错乱。

27. repo 从用户存储区(EEPRO)读数据到一个变量

仅增强型及以上系列适用

repo att,add

att:变量/常量

add: 用户存储区位置(从 0 开始)

实例 1:repo t0.txt,10 （从用户存储区的 10 位置读数据到 t0.txt 变量中,在储存区中的读取数据量为 t0.txt 的最大设置值+1，即 t0 的 txt-maxl 属性表示的大小+1)

实例 2.repo n0.val,10 （从用户存储区的 10 位置读数据到 n0.val,在存储区中的读取数据量为 4 字节)

深圳市淘晶驰电子有限公司 http://tjc1688.com/

地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

备注:
1.读入内容为变量字符串的时候，在储存区中的读取数据量为此变量的最大字符数+1。
2.读入内容为变量数值时候，在储存区中的读取数据量统一为 4 字节。
3.使用用户存储区读写操作过程中请切记规划好数据区位置，以免位置交错引起数据覆盖错乱。

28.wept 透传数据写入用户存储区(EEPROM)

仅增强型及以上系列适用

wept add,lenth
add: 用户存储区位置(从 0 开始)
lenth:透传长度
实例:wept 10,30 (透传 30 个字节的数据存到 EEPROM 的 10 位置，占用空间为 10-39)
备注:
1.发完透传指令后，用户需要等待设备响应才能开始透传数据，设备收到透传指令后，准备透传初始化数据大概需要 5ms 左右(如果在透传指令执行前串口缓冲区还有很多别的指令，那时间会更长)，设备透传初始化准备好以后会发送一个透传就绪的数据给用户（0XFE+结束符），表示设备已经准备好，此时可以开始发送透传数据。透传数据为纯 16 进制数据，不再使用字符串，也不再需要结束符,设备收完指定的数据量以后，才会恢复指令接收状态。否则一直处于数据透传状态，透传数据完成以后,设备会发送结束标记给用户（0XFD+结束符）。

29.rept 从用户存储区读取数据并透传发送到串口

仅增强型及以上系列适用

rept add,lenth
add: 用户存储区位置(从 0 开始)
lenth:读取并透传发送的长度
实例:rept 10,30 (从用户存储区的 10 位置读取 30 个字节并透传发送到串口)
备注:
不管存储区中的数据是字符串还是数值，设备都按 16 进制来读取和发送指定的字节数量到串口，并且不会发结束符。

30.cfgpio 扩展 IO 模式配置

仅增强型及以上系列适用

cfgpio id,state,obj
id:扩展 IO 的序号
state:配置模式(0-上拉输入模式,1-控件事件邦定输入模式,2-推挽输出模式,3-PWM 输出模式,4-开漏模式)
obj:绑定控件名称或 ID(此参数仅在配置为控件事件邦定输入模式下有效，其他模式下无效)
实例 1:cfgpio 0,0,0 (将 io0 配置为上拉输入，配置为此模式后，任意时刻可以使用系统变量 pio0 读取当前输入电平,如:n0.val=pio0)
实例 2:cfgpio 1,2,0 (将 io1 配置为推挽输出,配置为此模式后，任意时刻可以使用系统变量 pio1 控制当前输出电平，如:pio1=1)
实例 3:cfgpio 2,1,b0 (将 io2 配置为控件事件邦定输入，邦定控件为 b0,配置为此模式后，io2 产生下降沿的时候将触发 b0 控件的按下事件，产生上升沿的时候将触发 b0 控件的弹起事件)
实例 4:cfgpio 4,3,0 (将 io4 配置为 PWM 输出模式，配置之前需要先设置占空比，即系统变量变量中的 pwm4)
备注:
1.只有 io4-io7 才支持 PWM 输出，其他 IO 不支持。配置其他 IO 为 PWM 模式会报错。
2.使用控件事件邦定输入模式时，必须是在当前配置时刻的当前页面的控件才能邦定，不可以邦定其他页面的控件（即使是全局内存占用的控件也不可以），邦定当前页面控件以后，当重新刷新页面或者切换到别的页面后，

深圳市淘晶驰电子有限公司 http://tjc1688.com/
地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

邦定事件将不会继续触发，因此每次刷新页面需要重新邦定，建议将邦定代码写在页面的前初始化事件中最为合适。

分类二:GUI 绘图指令

注：GUI 绘图指令主要应用在如下场合：

当上位界面编辑软件无法实现您的某些特殊显示要求的时候，使用 GUI 指令自己绘图来实现自己想要的显示效果。大多数情况下其实是不需要使用这些绘图指令的，大多数的应用都可以通过界面编辑软件的控件操作来实现。

1.cls 清屏指令

cls color

color:十进制颜色值或颜色代号

实例 1:cls 1024 （用十进制 1024 的颜色值刷屏）

实例 2：cls RED （用代号为 RED 的颜色（RED 代表红色）刷屏）

备注：

1.想得到某个颜色的 10 进制数据可以使用设备配套的界面编辑软件”TJCHMI”获取，进入软件菜单栏”工具”-“取色工具”。

2.想了解设备支持的颜色代号表请参看本表格后面的”串口 HMI 颜色代号表”。

3.本指令表中所有指令中的颜色参数，全部都可以使用设备支持的颜色代号，也可以使用 10 进制的颜色值，请知晓。

2.pic 刷图指令

pic x,y,picid

x:起始点 x 坐标;

y:起始点 y 坐标;

picid:图片 ID;

实例 1:pic 10,20,0 （在坐标(10,20)位置显示资源文件中图片 ID 为 0 的图片）

实例 2:pic 40,50,1 （在坐标(40,50)位置显示资源文件中图片 ID 为 1 的图片 ）

3.picq 切图指令

格式:picq x,y,w,h,picid

x:屏幕起始点 x 坐标;

y:屏幕起始点 y 坐标;

w:区域宽度 ;

h:区域高度;

picid:图片 ID;

实例 1:picq 20,50,30,20,0 （将图片 0 起始坐标(0,0)宽度 30 高度 20 这个区域切到屏幕上显示，屏幕上的显示起始坐标为(20,50)）

备注：

此指令要求图片必须是全屏图片，否则切出来的图像不是你想要的。图片上的切图区域和屏幕上的显示区是重叠的。

4.xpic 高级切图指令

格式:xpic x,y,w,h,x0,y0,picid

x:屏幕起始点 x 坐标;

y:屏幕起始点 y 坐标;

w:区域宽度 ;

h:区域高度;
x0:图片起始点 x 坐标;
y0:图片起始点 y 坐标;
picid:图片 ID;
实例 1:xpic 20,50,30,20,40,15,0 （将图片 0 起始坐标(40,15)宽度 30 高度 20 这个区域切到屏幕上显示，屏幕上的显示起始坐标为(20,50)）

5. xstr 写字指令

xstr x,y,w,h,fontid,pointcolor,backcolor,xcenter,ycenter,sta,string
x:起始点坐标 x;
y:起始点坐标 y;
w:区域宽度;
h:区域高度;
fontid:字库 ID;
pointcolor:字体颜色;
backcolor:背景色(sta 设置为切图或图片时，backcolor 表示图片 ID);
xcenter:水平对齐方式(0 为左对齐，1 为居中，2 为右对齐);
ycenter: 垂直对齐方式(0 为上对齐，1 为居中，2 为下对齐);
sta:背景填充方式(0 为切图，1 为单色，2 为图片，3 为无背景,sta 设置为切图或图片时，backcolor 表示图片 ID)
string:字符内容;
实例 1:xstr 0,0,100,30,1,RED,BLACK,1,1,1,”中国”
实例解释：使用字库 1 在起始坐标(0,0)，宽度 100,高度 30 这个区域写出”中国”，字体色为 RED，背景色为 BLACK(如果不想写背景色(即无背景)可以设置 sta 参数为 3),水平对齐方式为居中，垂直对齐方式也为居中。
备注：
1.字符写到超过设定的 w 以后将自动换行，如果换行到 h 之后还有剩下的字符没写完，将会被忽略。
2.关于颜色值的说明请参看 cls 指令的备注。

6. fill 区域填充指令

fill x,y,w,h,color
x:起始点坐标 x ;
y:起始点坐标 y;
w:区域宽度;
h:区域高度;
color:填充颜色;
实例 1:fill 0,0,100,30,RED （在起始坐标(0,0)宽度 100，高度 30 这个区域填充 RED 颜色）
备注：
关于颜色值的说明请参看 cls 指令的备注。

7. line 画线指令

line x,y,x2,y2,color
x:起始点坐标 x;
y:起始点坐标 y;
x2:结束点坐标 x ;
y2:结束点坐标 y;

color:画线颜色；
实例 1:line 0,0,100,100,RED （在坐标(0,0)和坐标(100,100)之间画出一条 RED 颜色的线）
备注：
关于颜色值的说明请参看 cls 指令的备注。

8. draw	画矩形

draw x,y,x2,y2,color
x:起始点坐标 x ；
y:起始点坐标 y；
x2:结束点坐标 x ；
y2:结束点坐标 y；
color:画线颜色；
实例 1:draw 0,0,100,100,RED （画一个矩形，左上角为(0,0),右下角为(100,100)， 颜色为 RED）
备注：
1. draw 画出来的是空心矩形，需要填充实心矩形的话请直接使用 fill 区域填充指令。
2. 关于颜色值的说明请参看 cls 指令的备注。

9. cir	画空心圆

cir x,y,r,color
x:圆心坐标 x
y:圆心坐标 y
r:半径
color:画线颜色；
实例 1:cir 100,100,30,RED 以坐标(100,100)为圆心画一个半径为 30 的空心圆，颜色为 RED
备注：
关于颜色值的说明请参看 cls 指令的备注。

10. cirs	画实心圆

cirs x,y,r,color
x:圆心坐标 x
y:圆心坐标 y
r:半径
color:填充颜色；
实例 1:cir 100,100,30,RED 以坐标(100,100)为圆心画一个半径为 30 的实心圆，填充颜色为 RED
备注：
关于颜色值的说明请参看 cls 指令的备注。

提示：本指令表中所有指令中的颜色参数，全部都可以使用设备支持的颜色代号，也可以使用 10 进制的颜色值，请知晓，想了解设备支持的颜色代号表请参看本表格后面的”串口 HMI 颜色代号表”，想得到某个颜色的 10 进制数据可以使用设备配套的界面编辑软件”USART HMI”获取，进入软件菜单栏”工具”-“取色工具”。

串口 HMI 语句

注：所有的逻辑语句只能在上位编辑状态下写入控件的事件中，不支持串口传输逻辑语句。

1.if

实例 1: （如果 t0.txt 等于”123456”那么就切换到页面 1）

```
if(t0.txt==”123456”)
{
page 1
}
```

实例 2:(以下语句写在 b0 按钮的按下事件中将实现 b0 的 txt 内容在开始和停止之间来回切换)

```
if(b0.txt==”开始”)
{
b0.txt=”停止”
}else
{
b0.txt=”开始”
}
```

实例 3:(以下语句写在 b0 按钮的按下事件中将实现 b0 的 txt 内容在 1,2,3 之间来回切换)

```
if(b0.txt==”1”)
{
b0.txt=”2”
}else if(b0.txt==”2”)
{
b0.txt=”3”
}else
{
b0.txt=”1”
}
```

备注:

1.数值类型变量支持：1.大于判断(>) 2.小于判断(<) 3.等于判断(==) 4.不等于判断(!=) 5.大于等于判断(>=)。6.小于等于判断(<=)。

2.字符串类型仅支持 1.等于判断(==)　2.不等于判断(!=)。

2.while

实例 1: (n0.val 一直自加到 100 为止，在自加过程中屏幕不会刷新显示，直到整个过程所有语句结束)

```
while(n0.val<100)
{
n0.val++
}
```

实例 2: (n0.val 一直自加到 100 为止，在自加过程中屏幕会一直不断的刷新 n0 控件的显示)

```
while(n0.val<100)
{
```

```
n0.val++
doevents
}
```

备注:

1.在一个较多指令的过程执行中，或者在一个较长时间的循环语句中，系统所有控制权被此过程全部占用，在过程结束之前，尽管相应的内存数据可以任意正常读写，但是屏幕不会刷新显示，加入 doevents 后可以转让控制权给屏幕刷新，执行 doevents 之后，屏幕会刷新所有被改变过的控件，刷新完之后，控制权交回当前过程继续执行。防止屏幕呈现假死的显示状态。

2.while 语句循环过程中，设备不会响应触摸事件，串口指令会接收到缓冲区，但不会执行，直到当前过程所有语句执行完毕为止，请慎重使用，以防进入死循环。

3. for

实例 1: (n0.val 一直自加到 100 为止，在自加过程中屏幕不会刷新显示，直到整个过程所有语句结束)

```
for(n0.val=0;n0.val<100;n0.val++)
{
}
```

实例 2: (n0.val 一直自加到 100 为止，在自加过程中屏幕会一直不断的刷新 n0 控件的显示)

```
for(n0.val=0;n0.val<100;n0.val++)
{
doevents
}
```

备注:

1.在一个较多指令的过程执行中，或者在一个较长时间的循环语句中，系统所有控制权被此过程全部占用，在过程结束之前，尽管相应的内存数据可以任意正常读写，但是屏幕不会刷新显示，加入 doevents 后可以转让控制权给屏幕刷新，执行 doevents 之后，屏幕会刷新所有被改变过的控件，刷新完之后，控制权交回当前过程继续执行。防止屏幕呈现假死的显示状态。

2.for 语句循环过程中，设备不会响应触摸事件，串口指令会接收到缓冲区，但不会执行，直到当前过程所有语句执行完毕为止，请慎重使用，以防进入死循环。

串口 HMI 系统变量列表

注：所有变量名称使用小写字符

序号	名称	含义	示例/备注
1	dp	当前页面 ID	1. dp=1 （设置当前页面为 1,等同于 page 1） 2. print dp(发送当面页面 ID 到串口) 3. n0.val=dp （当前页面 ID 赋值给 n0.val)
2	dim	当前背光亮度值(0-100)	1.dim=50 2.dim=dim+10 3.dim=dim-10
3	dims	上电默认背光亮度值(0-100)	1.dims=50 2.dims=dims+10 3.dims=dims-10
4	baud	当前波特率值	baud=9600 备注: 设备支持的波特率有:2400 4800 9600 19200 38400 57600 115200
5	bauds	上电默认波特率值	bauds=9600
6	spax	字符显示横向间距(上电默认为 0)	spax=2 备注： 仅对 xstr 指令写出来的字符有效，控件带的字符显示间距由控件内部的属性决定。
7	spay	字符显示纵向间距(上电默认为 0)	spay=2 备注： 仅对 xstr 指令写出来的字符有效，控件带的字符显示间距由控件内部的属性决定。
8	thc	触摸绘图时的画笔色	1.thc=RED 2.thc=1024
9	thdra	触摸绘图功能开关	thdra=0 （关闭） thdra=1(打开)
10	ussp	无串口数据自动睡眠时间(单位:秒,最小 3，最大 65535，上电默认 0)	ussp=30(30 秒无串口数据自动进入睡眠模式)
11	thsp	无触摸操作自动睡眠时间(单位:秒,最小 3，最大 65535，上电默认 0)	thsp=30(30 秒无触摸操作自动进入睡眠模式)
12	thup	睡眠模式下触摸自动唤醒开关(上电默认 0)	thup=0(睡眠后触摸不会自动唤醒) thup=1(睡眠后触摸自动唤醒) 备注: 不管 thup 为 0 还是 1，睡眠模式下有触摸操作的时候设备均会发送触摸坐标到串口。
13	usup	睡眠模式下串口数据自动唤醒开关(上电默认 0)	usup=0(睡眠后串口不会自动唤醒) usup=1(睡眠后串口自动唤醒) 备注: 上电默认为 0，不会自定唤醒，需要发送 sleep=0 才能唤醒屏幕，如果设置为 1，串口收到任何数据都会立刻自动唤醒。
14	wup	睡眠唤醒后刷新页面设置	wup=255(上电默认，睡眠唤醒后刷新睡眠前页面) wup=2 (睡眠唤醒后刷新页面指定页面:2)

深圳市淘晶驰电子有限公司 http://tjc1688.com/
地址：深圳市龙华新区大浪华盛路 45 号宝华诚工业园 3 栋 2 楼 C 区 电话：0755-21004082

			备注： 设备已经在睡眠状态下，也可以执行串口传过来的 wup=X 赋值。
15	sleep	睡眠	sleep=0　（退出睡眠） sleep=1　（进入睡眠） 备注： 睡眠状态下可以执行如下指令：get,print,printh。 也可以 执行 sleep=1，wup=X 的赋值语句，并且支持上位软件联机，其他指令不会执行。如果是增强型及以上系列并且扩展 IO 配置为绑定控件事件时，睡眠模式下也不会产生中断事件。
16	bkcmd	设置串口指令执行成功或者失败的数据返回(上电默认为 2)	bkcmd=0(不返回结果) bkcmd=1(只返回成功的结果) bkcmd=2(只返回失败的结果) bkcmd=3(成功或者失败都返回结果) 备注: 此设置只影响串口指令执行成功或者失败的结果返回，上位软件编辑界面时写入的指令执行错误的时候一定会返回错误结果，成功的时候一定不会返回执行结果。此设置也不会影响获取设备控件数据时的数据返回。
17	sendxy	实时发送触摸坐标功能开关	sendxy=0(关闭)　sendxy=1(打开) 备注: 1 打开发送功能以后,有触摸按下的时候设备会通过串口发送触摸坐标。 2. 发送坐标的格式请参看本表格后面的”串口 HMI 设备返回数据格式”表格。
18	delay	延时	delay=100　（让设备停顿 100ms） 备注： 执行延时指令后，设备 CPU 不会执行任何指令，但是会继续接受串口指令保存到串口指令缓存区。
19	rand	随机数	dim=rand （把一个随机数赋值给背光亮度） n0.val=rand (把一个随机数赋值给 n0.val 变量) 备注： 1. 使用随机数之前需要先使用 randset 指令设定一次随机数产生范围，如果不设置，默认是最小 0，最大 2147483647。设置完范围以后，每读取一次系统变量 rand 将会得到一个随机数。 2. 使用 randset 指令每设定一次范围，将一直有效，直到重新上电或者设备复位才会恢复默认。

20	sys0 sys1 sys2	内置数值变量	sys0=10 sys1=40 sys2=60 n.val=sys2 备注： sys0,sys1,sys2 三个数值变量为全局类型，不用定义，不用创建，任何页面任何时刻任意使用。上电默认为 0，可以读取，可以赋值，数据类型为 int 类型(即：最小-2147483648, 最大 2147483647)。页面间传递数值的时候推荐使用。使用内置数值变量做运算解析速度比使用控件属性变量更快。
21	tch0-tch3	实时触摸坐标	tch0:当前触摸坐标 X tch1:当前触摸坐标 Y tch2:上一次按下时的坐标 X tch3:上一次按下时的坐标 y 备注： 触摸坐标只能读取，不能赋值，没有按下时，实时坐标数据为 0。
22	addr	设备地址	字符串写法:addr=256 HEX 写法:addr=0x0100 以上两条写法是同一个意思，配置的是同一个地址，配置之后有断电保存功能。 备注： 1. 有效地址范围为 256-2815 (即 0x0100-0x0aff),0 为无地址,65535 为广播地址，广播地址只能用于广播数据，不能配置某个设备为广播地址，出厂默认地址为 0,即没有地址。 2.向一个有地址的设备发送指令时，需要在指令前加上 2 字节的地址数据，以 hex 方式发送,2 字节小端模式,比如设备配置的地址为 addr=256,那么发送给他指令时需要在指令前面增加两个字节:0x00 0x01(注意，配置的时候是 0x0100,发送指令的时候是低位在前，所以是 0x00 0x01 跟配置的写法是相反的)。
23	rtc0-rtc6	RTC 时钟变量 **(仅增强型及以上系列适用)**	n0.val=rtc5 (当前 RTC 的秒数值赋值给 n0.val) rtc0=2016(RTC 的年设置为 2016) cov rtc5,t0.txt,0(当前 RTC 的秒数值转换给 t0.txt) 备注： rtc0-rtc6 分别表示年，月，日，时，分，秒,星期。
24	pio0-pio7	扩展 IO 端口 **(仅增强型及以上系列适用)**	pio4=1 (IO4 置为 1) n0.val=pio2 (io2 的电平状态赋值给 n0.val)

			cov pio3,t0.txt,0 （io3 的电平状态转换给 t0.txt） 备注： 1.使用pio端口之前一定要先使用cfgpio指令配置好 IO 模式。 2.上电默认所有扩展 IO 模式为上拉输入(内部上拉电阻为 50K)。
25	pwm4-pwm7	扩展 IO 占空比 **(仅增强型及以上系列适用)**	pwm4=30 （设置 pwm4 占空比为 30） pwm5=90 （设置 pwm5 占空比为 90） 备注： 1. 占空比最小值 0 最大值 100,上电默认 50。 2.pwm4-pwm7 依次对应扩展 IO 中的 io4-io7, io0-io3 不支持 PWM 输出。 3.设置好 PWM 占空比以后,需要使用 cfgpio 指令配置此 IO 的模式为 PWM 输出模式，相应 IO 才会开始输出 PWM,配置完 PWM 模式后在 PWM 输出的过程中可以随时修改占空比，不用重新配置。 4. 上电默认所有扩展 IO 模式为上拉输入（内部上拉电阻为 50K)。
26	pwmf	PWM 输出的频率 **(仅增强型及以上系列适用)**	pwmf=1024 （设置 pwm 的频率为 1024HZ） n0.val=pwmf （将 PWM 频率赋值给 n0.val） cov pwmf,t0.txt(将 PWM 频率转换给 t0.txt) 备注： 1.频率单位为:HZ,范围为最小 1，最大 65535HZ,上电默认 1000HZ。 2.所有 PWM 输出统一为一个频率,不可单独设置。

串口 HMI 颜色代号表

注：所有代号的书写均为大写

代号	10 进制	所表示的颜色
RED	63488	红色
BLUE	31	蓝色
GRAY	33840	灰色
BLACK	0	黑色
WHITE	65535	白色
GREEN	2016	绿色
BROWN	48192	橙色
YELLOW	65504	黄色

串口 HMI 设备返回数据格式

表格一：串口指令执行成功或失败的通知格式

1. 只有当系统变量 bkcmd 为非 0 的时候才会返回指令执行成功或者失败数据，每次上电后 bmcmd 默认为 2，即只返回指令执行出错的结果。
2. 上位软件编辑时写进资源文件的代码不受 bkcmd 影响，执行有错误时一定会返回错误数据，成功时不返回数据。
3. 设备返回数据的结束符为”0XFF 0XFF 0XFF”三个字节。

返回数据第一位	含义	格式
0X00	无效指令	0X00+结束符 (当收到用户发来的无效指令时返回此数据)
0X01	指令成功执行	0X01+结束符 (用户发来的指令被成功执行完毕时返回此数据)
0X02	控件 ID 无效	0X02+结束符 (用户发来的指令中包含无效控件 ID 或者无效控件名称时返回此数据)
0X03	页面 ID 无效	0X03+结束符 (用户发来的指令中包含无效页面 ID 或者无效页面名称时返回此数据)
0X04	图片 ID 无效	0X04+结束符 (用户发来的指令中包含无效图片 ID 时返回此数据)
0X05	字库 ID 无效	0X05+结束符 (用户发来的指令中包含无效字库 ID 时返回此数据)
0X11	波特率设置无效	0X11+结束符 (用户发来的波特率配置指令中包含无效波特率参数) 设备支持的波特率有:2400 4800 9600 19200 38400 57600 115200
0X12	曲线控件 ID 号或通道号无效	0X12+结束符 (用户使用 add 指令往曲线控件添加数据的时候，曲线控件 ID 号或通道号无效时返回此数据)
0X1A	变量名称无效	0X1A+结束符 当串口收到的变量名称为无效名称时返回此数据 注：控件属性也称为变量，比如您设置一个控件的属性的时候，输入的是一个它没有的属性名称，也会返回此数据。
0X1B	变量运算无效	0X1B+结束符 比如文本控件 t0 的 txt 属性赋值时应该写成 t0.txt=”abc”如果你写成 t0.txt=abc 就出错了，再比如进度条 j0 的 val 属性应该是数值，所以要写成 j0.val=50, 如果写成 j0.val=”50”或者 j0.val=abc 也会出错

0X1C	赋值操作失败	0X1C+结束符 属性赋值失败的时候返回此数据
0X1D	EEPROM 操作失败	0X1D+结束符 操作 EEPROM 失败时返回此数据
0X1E	参数数量无效	0X1E+结束符 用户输入的指令中参数数量错误的时候返回此数据
0X1F	IO 操作失败	0X1F+结束符 操作 IO 失败时返回此数据
0X20	转义字符使用错误	0X20+结束符 转义字符使用错误时返回此数据
0X23	变量名称太长	0X23+结束符 变量名称长度最大 29 个字符，超出就会返回此数据
0X24	串口缓冲区溢出	0X24+结束符 当串口缓冲区被占满以后会返回此数据 (缓冲区溢出以后，缓冲队列里的指令执行完成后会为缓冲区腾出空间以继续接收指令，在此之前，串口收到的数据将会丢弃)

表格二：其他数据返回格式

1. 设备返回数据的结束符为”0XFF 0XFF 0XFF”三个字节。

2. 以下数据的返回不受 bkcmd 影响。

返回数据第一位	含义	格式
0X65	触摸热区事件返回	0X65+页面 ID+按键 ID+触摸事件+结束符 (用户创建的控件被按下或弹起时返回此数据，前提是您勾选了控件的”发送键值”选框) (触摸事件的定义：按下事件 0x01 弹起事件 0X00) 举例:0X65 0X00 0X02 0X01 0XFF 0XFF 0XFF 含义:页面 0 按钮 2 按下
0X66	当前页面的 ID 号返回	0X66+页面 ID+结束符 (设备收到”sendme”指令时会返回此数据) 举例：0X66 0X02 0XFF 0XFF 0XFF 含义：当前页面 ID 为 2
0X67	触摸坐标数据返回	0X67++坐标 X 高位+坐标 X 低位+坐标 Y 高位+坐标 Y 低位+触摸事件状态+结束符 (当系统变量”sendxy”为 1 之后，有触摸事件时返回此数据) (触摸事件的定义：按下事件 0x01 弹起事件 0X00) 举例:0X67 0X00 0X7A 0X00 0X1E 0X01 0XFF 0XFF 0XFF 含义:坐标(122,30) 事件：按下
0X68	睡眠模式触摸事件	0X68++坐标 X 高位+坐标 X 低位+坐标 Y 高位+坐标 Y 低位+触摸事件状态+结束符 (当设备进入睡眠模式后，有触摸事件时返回此数据) (触摸事件的定义：按下事件 0x01 弹起事件 0X00)

		举例:0X68 0X00 0X7A 0X00 0X1E 0X01 0XFF 0XFF 0XFF 含义:坐标(122,30) 事件: 按下
0X70	字符串变量数据返回	0X70+变量内容 ASCII 码+结束符 使用 get 指令获取的变量为字符串类型时，返回此数据. 举例:0X70 0X61 0X62 0X63 0XFF 0XFF 0XFF 含义：返回字符串数据:”abc”
0X71	数值变量数据返回	0X71+变量二进制数据（4 字节小端模式,低位在前）+结束符 使用 get 指令获取的变量为数值时，返回此数据. 举例:0X71 0X66 0X00 0X00 0X00 0XFF 0XFF 0XFF 含义：返回数值数据:102
0X86	设备自动进入睡眠模式	0x86+结束符 只有设备自动进入睡眠模式的时候会返回此数据，如果是执行串口指令 sleep=1 进入的睡眠不会返回此数据
0X87	设备自动唤醒	0x87+结束符 只有设备自动唤醒的时候会返回此数据，如果是执行串口指令 sleep=0 唤醒的睡眠不会返回此数据
0X88	系统启动成功	0x88+结束符 设备上电初始化成功之后发送此数据
0X89	开始 SD 卡升级	0x89+结束符 设备上电检测到 SD 卡之后将发送此数据,然后进入升级界面
0XFD	透传数据完成	0xFD+结束符 透传数据结束并处理数据完成后发送此数据
0XFE	数据透传就绪	设备收到数据透传指令后，进入透传数据初始化，初始化完成以后发送此数据，表示此时已经进入数据透传模式，可以开始数据透传