串口指令集

发布版本 V1.2

目录

1.串口指令概述

◆ 指令格式说明：

1. 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。
2. 所有指令名使用小写字母(仅代表指令名称为小写，与参数大小状态无关)。

◆ 颜色格式：

本液晶模块采用 16 位真彩色显示，画面细腻，色彩丰富，数据格式采用标准 565 颜色格式：

65536 色的设置方法:

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
65536 色	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

举例说明：纯红色=F800(16 进制)/63488(十进制)，纯蓝色= 001F(16 进制)/ 31(十进制)

注：用户可以通过软件调试并获取期望的颜色值，颜色值只能用十进制的数值表示；

◆ 操作指令：

指令集分两部分：组态控件指令集和基本指令集。

两者主要区别是：组态指令集直接是面向对象操作，而这些对象的相关参数全部预先在上位机软件中进行了配置，与图片一起下载到了屏的存储器中；基本指令集可以理解为最底层的指令集，大部分操作必须包含坐标、颜色、字体等参数信息。当上位机界面编辑软件无法实现您的某些特殊显示要求的时候，用户使用基本指令来实现自己想要的显示效果，例如清屏。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。

◆ 说明：

指令结束符为“0x0D 0x0A”两个字节，使用 FY HMI 系统发送指令时，自动追加结束符；使用通用型串口助手时，确保串口助手能自动追加“0x0D 0x0A”，否则需手动追加“0x0D 0x0A”。

2.基本指令集

指令	功能	参数	实例	说明
cls	清屏	数量：1 参数 1：十进制颜色值	cls 1024（用十进制 1024 的颜色值刷屏）	1.想得到某个颜色的 10 进制数据可以进入软件菜单栏”工具”-“取色工具”。 2.本指令表中所有指令中的颜色参数，全部使用 10 进制的颜色值
pic	刷图	数量：3 参数 1：起始点 x 坐标 参数 2：起始点 y 坐标 参数 3：图片 ID	1.pic 50,50,0（在坐标(50,50)位置显示资源文件中图片 ID 为 0 的图片）	
xpic	高级切图	数量：7 参数 1：屏幕起始点 x 坐标 参数 2：屏幕起始点 y 坐标 参数 3：区域宽度 参数 4：区域高度 参数 5：图片起始点 x 坐标 参数 6：图片起始点 x 坐标 参数 7：图片 ID	1. xpic 50,50,30,20,30,30,0 （将图片 0 起始坐标(30,30)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(50,50)）	
picq	切图	数量：5 参数 1：屏幕起始点 x 坐标 参数 2：屏幕起始点 y 坐标	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)）	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。

		参数 3：区域宽度 参数 4：区域高度 参数 5：图片 ID		
wristr	写字指令	数量：11 参数 1：起始点 x 坐标 参数 2：起始点 y 坐标 参数 3：区域宽度 参数 4：区域高度 参数 5：字库 ID 参数 6：字体颜色 参数 7：背景色(sta 设置为切图或图片时，backcolor 表示图片 ID)； 参数 8：水平对齐方式(0 为左对齐，1 为居中，2 为右对齐)； 参数 9：垂直对齐方式(0 为上对齐，1 为居中，2 为下对齐)； 参数 10：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) 参数 11：字符内容	1. wristr 0,0,200,40,0,63488,1024,1,1,1,"FY HMI"(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 63488，背景色为 31(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。)	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。
fill	区域填充	数量：5 参数 1：起始点 x 坐标 参数 2：起始点 y 坐标	1.fill 0,0,100,30,1024 (在起始坐标(0,0)宽度 100，高度 30 这个区域填充 颜色值为 1024 的颜色)	关于颜色值的说明请参看 cls 指令的备注

		参数 3：区域宽度 参数 4：区域高度 参数 5：填充颜色		
line	画线	数量：5 参数 1：起始点 x 坐标 参数 2：起始点 y 坐标 参数 3：结束点 x 坐标 参数 4：结束点 y 坐标 参数 5：画线颜色值	1. line 0,0,100,100,1024 （在坐标(0,0)和坐标(100,100)之间画出一条 颜色值为 1024 颜色的线）	关于颜色值的说明请参看 cls 指令的备注。
draw	画矩形	数量：5 参数 1：起始点 x 坐标 参数 2：起始点 y 坐标 参数 3：结束点 x 坐标 参数 4：结束点 y 坐标 参数 5：画线颜色值	1. draw 0,0,100,100,1024 （画一个矩形，左上角为(0,0),右下角为(100,100)， 颜色为 RED）	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
cir	画空心圆	数量：4 参数 1：圆心 x 坐标 参数 2：圆心 y 坐标 参数 3：圆半径 参数 4：画线颜色值	1. cir 100,100,30,1024 （以坐标(100,100)为圆心画一个半径为 30 的空心圆，颜色值为 1024）	关于颜色值的说明请参看 cls 指令的备注。

solcir	画实心圆	数量：4 参数 1：圆心 x 坐标 参数 2：圆心 y 坐标 参数 3：圆半径 参数 4：画线颜色值	1. solcir 100,100,30,1024（以坐标(100,100)为圆心画一个半径为 30 的空心圆，颜色值为 1024）	关于颜色值的说明请参看 cls 指令的备注。

3.组态控件指令集

指令	功能	参数	实例	说明
page	刷新页面	数量：1 个 类型：页面 ID 或页面名称实例	1.page 0(刷新 ID 为 0 的页面) 2.page main(刷新名称为 main 的页面)	当需要跳转到其他界面时，可使用 page 指令实现
ref	重绘控件	数量：1 个 类型：控件 ID 或控件名称实例	1.ref 1(重绘 ID 为 1 的控件) 2.ref textBox0(重绘名称为 textBox0 的控件)	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，可以使用 ref 来重绘。
click	按下/弹起事件	数量：2 个 类型 1：控件 ID 或控件名称实例 类型 2：事件类型，0 为弹起，1 为按下	1.click button0,0 （激活名称为 btn0 的控件的弹起事件） 2.click button0,1 （激活名称为 btn1 的控件的按下事件）	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。
get	获取值	数量：1 个 类型：变量名称	1.get textBox0.txt （返回控件 textBox0 的 txt 属性值） 2.get number0.val(返回控件 number0 的 val 属性值)	
read	获取变量值/常量值	数量：1 个 类型 1：变量名称	1.read textBox0.txt （返回控件 textBox0 的 txt 属性值）	1.使用 read 指令获取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

			2. read number0.val(返回控件 number0 的 val 属性值) 3. read “123”(返回常量字符串”123”即：0x31 0x32 0x33) 4. read 123(返回常量数值：123 即：0x7b 0x00 0x00 0x00)	回变量的 4 字节十六进制数据(int 类型)，数值的存放模式为小端模式（即低位在前，高位在后)。 2.使用 read 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. read 指令可以配合 readh 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)。
readh	获取自定义 16 进制 byte	数量：1 个 类型：需要发送的字符的 16 进制字符串表达式	1.readh d0 a0(让设备发送 0xd0 0xa0 两个字节)	1.使用 readh 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。
vis	隐藏/显示控件	数量：2 个 类型 1：控件名称或控件 ID 类型 1：状态(0 为隐藏，1 为显示)	1.vis button0,0(隐藏 button0 控件) 2.vis button0,1(显示 button0 控件)	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)
tsw	控件触摸使能	数量：2 个 类型 1：控件名称或控件 ID 类型 1：状态(0 为隐藏，1 为显示)	1.tsw button0,0 (让名称为 button0 的控件触摸失效) 2.tsw button0,1 (让名称为 button0 的控件触摸有效)	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)。
randset	随机数范围设置	数量：2 个 类型 1：最小值 类型 2：最大值	1.randset 0,100 (设置当前随机数产生范围为最小 0，最大 100)	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)。

add	往曲线控件添加数据	数量：3 参数 1：曲线控件 ID 序号 参数 2：曲线控件通道号 参数 3：（最大 255，最小 0）	1. add 1,0,60（往 ID 为 1 的曲线控件的 0 通道添加数据 60）	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。
addw	往曲线控件添加指定波形数据	数量：5 参数 1：曲线控件 ID 序号 参数 2：曲线控件通道号 参数 3：波形编号(0-2) 参数 4：曲线波形幅度 参数 5：曲线波形偏置	1. addw 1,0,0,100,50（往 ID 为 1 的曲线控件的 0 通道添加幅度为 100，偏置为 50 的正弦波数据）	1. 幅度值与偏置值只支持 8 位数据，最小 0，最大 255。 2. 波形编号共有三个：0-正弦波；1-方波；2：三角波；
cle	清除曲线控件数据	数量：2 参数 1：曲线控件 ID 序号 参数 2：曲线控件通道号	1. cle 1,0(清除 ID 为 1 的曲线控件的 0 通道数据） 2. cle 1,255(清除 ID 为 1 的曲线控件的所有通道数据）	1. 通道号为 255 时表示清除此曲线控件内的所有通道数据。
addt	曲线数据透传指令	数量：3 参数 1：曲线控件 ID 序号 参数 2：曲线控件通道号 参数 3：本次透传数据的点数量	1. 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）	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。
doevents	转让系统控制权给屏幕刷新	无	1.doevents	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
repageid	获取当前页面 ID 号到串口	无	1.repageid	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 repageid 语句即可。
covx	变量类型转换	数量：4 参数 1：源变量 参数 2：目标变量 参数 3：字符串的长度(0 为自动长度，非 0 为固定长度) 参数 4：申明数值类型(0-数字;1-货币;2-Hex)	1.covx slider0.val,textBox0.txt,0,0(把滑块 slider0 的 val 数值变量转换成 10 进制数字子符串并赋值给文本 textBox0 的 txt 变量,长度为自动) 2. covx textBox0.txt, slider0.val,0,0(把文本 textBox0 的 txt 十进制数字字符串变量转换为数值并赋值给滑块 slider0 的 val 数值变量,长度为自动)	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。

strlen	字符串变量字符长度测试	数量：2 参数 1:需要测试的字符串变量 参数 2:把测试结果赋值给此变量	1. strlen textBox0.txt,n0.val （把字符串变量 textBox0.txt 的实际字符长度赋值给 n0.val）	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。
btlen	字符串变量字节长度测试	数量：2 参数 1:需要测试的字符串变量 参数 2:把测试结果赋值给此变量	1. btlen textBox0.txt,number0.val （把字符串变量 textBox0.txt 的实际字节长度赋值给 n0.val）	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。
substr	字符串截取	数量：4 参数 1：源变量 参数 2：目标变量 参数 3：在源变量中的字符起始位置 参数 4：截取字符串长度实例	1. substr textBox0.txt,t1.txt,0,2 （从 textBox0.txt 中的 0 位置开始截取 2 个字符赋值给 t1.txt）	
touch_j	触摸校准	无	touch_j （进入触摸校准功能）	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。
ref_stop	暂停屏幕刷新	无	ref_stop	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。

ref_star	恢复屏幕刷新	无	ref_star	此指令和 ref_stop 配合使用。
com_stop	暂停串口指令执行	无	com_stop	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。
com_star	恢复串口指令执行	无	com_star	
code_c	清空串口指令缓冲区中还没有执行的所有指令	无	code_c	
rest	复位	无	rest	

4.串口 HMI 语句

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 。

语句	实例	说明

if	1. 如果系统变量 sys0 等于”100”那么就刷新页面 0
	if(sys0 ==100)
	{
	page 0
	}
		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(!=)。

while	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(number0.val<100)
	{
	number0.val++
	}
	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(number0.val<100)
	{
	number0.val++
	doevents
	}
		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。

for	1. number0.val 每次加一，循环 100 次，在循环过程中屏幕不会刷新显示，直到循环过程所有语句结束，才有刷新屏幕显示	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

	for(number0.val=0; number0.val<100; number0.val++)
	{
	prog0.val= number0.val
	}
	2. var0.val 每次加一，循环 100 次，在循环过程中屏幕会不断的刷新 prog0 控件的显示
	for(var0.val=0;var0.val<100;var0.val++)
	{
	doevents
	prog0.val=var0.val
	}
		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。

5.串口 HMI 系统变量列表

所有变量名称使用小写字符

指令	功能	实例	备注
dp	当前页面 ID	1. dp=1（设置当前页面为 1,等同于 page 1） 2. read dp(发送当面页面 ID 到串口) 3. number0.val=dp（当前页面 ID 赋值给 number0.val	
presdim	当前背光亮度值(0-100)	1. presdim=50 2. presdim=presdim +10	
defdim	上电默认背光亮度值(0-100)	1.defdim=50	

		2. defdim=defdim +10 3. defdim=defdim -10	
presbaud	当前波特率值	presbaud=9600	设备支持的波特率有 :9600 19200 38400 57600 115200 230400 460800
defbaud	上电默认波特率值	defbaud=9600	
spax	字符显示横向间距(上电默认为 0)	spax=2	仅对 wristr 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。
spay	字符显示纵向间距(上电默认为 0)	spay=2	仅对 wristr 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
thc	触摸绘图时的画笔色	1. thc=1024	
thdra	触摸绘图功能开关	1. thdra=0 (关闭) 2. thdra=1(打开)	
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)	
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)	
thup	睡眠模式下触摸自动唤醒开关(上电默认 0)	1. thup=0(睡眠后触摸不会自动唤醒) 2. thup=1(睡眠后触摸自动唤醒)	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
usup	睡眠模式下串口数据自动唤醒开关(上电默认 0)	1. usup=0(睡眠后串口不会自动唤醒) 2. usup=1(睡眠后串口自动唤醒)	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
wup	睡眠唤醒后刷新页面设置	1. wup=255(上电默认，睡眠唤醒后刷新睡眠前页面)	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。

		2.wup=2 (睡眠唤醒后刷新页面指定页面:2)	
sleep	睡眠	1.sleep=0 (退出睡眠) 2.sleep=1 (进入睡眠)	睡眠状态下可以执行如下指令：get,read,readh。 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。
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)	1.bkcmd=0(不返回结果) 2.bkcmd=1(只返回成功的结果) 3.bkcmd=2(只返回失败的结果) 4.bkcmd=3(成功或者失败都返回结果)	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。
rexy	实时获取触摸坐标功能开关	1.rexy=0(关闭) 2.rexy=1(打开)	打开发送功能以后，有触摸按下的时候设备会通过串口获取触摸坐标。
delay	延时	delay=100 (让设备停顿 100ms)	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。
rand	随机数	1.presdim=rand (把一个随机数赋值给背光亮度) 2.number0.val=rand (把一个随机数赋值给 number0.val 变量)	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。
sys0 sys1 sys2	内置数值变量	1.sys0=10 2.sys1=40 3.sys2=60 4.number0.val=sys2	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。

tch0-tch3	触摸坐标	1. get tch0:当前触摸坐标 X 2. get tch1:当前触摸坐标 Y 3. get tch2:上一次按下时的坐标 X 4. get tch3:上一次按下时的坐标 y	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。
addr	设备地址	1. addr=1（设置设备地址为 1） 2. addr=0x01（设置设备地址为 1）	1. 有效地址范围为 1-254(即 0x01-0xfe),0 为无地址,255 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，需要在指令前加上地址数据,发送格式为地址数据：指令数据，例：1：ref 0 （字符串发送）或 31 3A 72 65 66 20 30 0D 0A（HEX 发送，0D 0A 为结束符） 3. 配置以后有断电保存功能