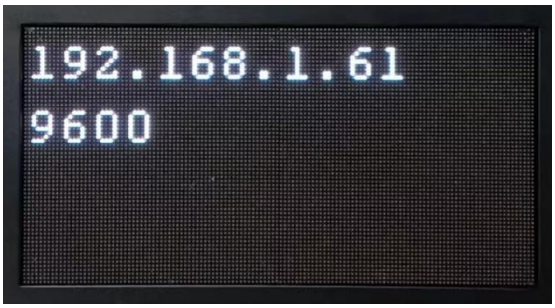
ZSWE-LED102A 物联网报警显示屏
脚本编程手册

咨询电话：18181995160 李工

上电信息说明



设备上电后，会自动显示本机以太网接口的IP地址，以及当前的串口波特率信息。

注：参数配置接线说明：

温馨提示：配置参数前，用户可通过一根USB转RS485或RS232串口线配件，用于连接电脑和设备的485或232，并在电脑安装驱动软件识别USB驱动，设备管理器识别出COM口后可与设备建立通讯，进行参数配置。

用户也可以选择通过以太网进行参数配置，将设备的网口通过网线连接到路由器，设备默认出厂IP地址为192.168.1.61，配置软件可以通过TCP连接到这个IP地址进行参数配置。

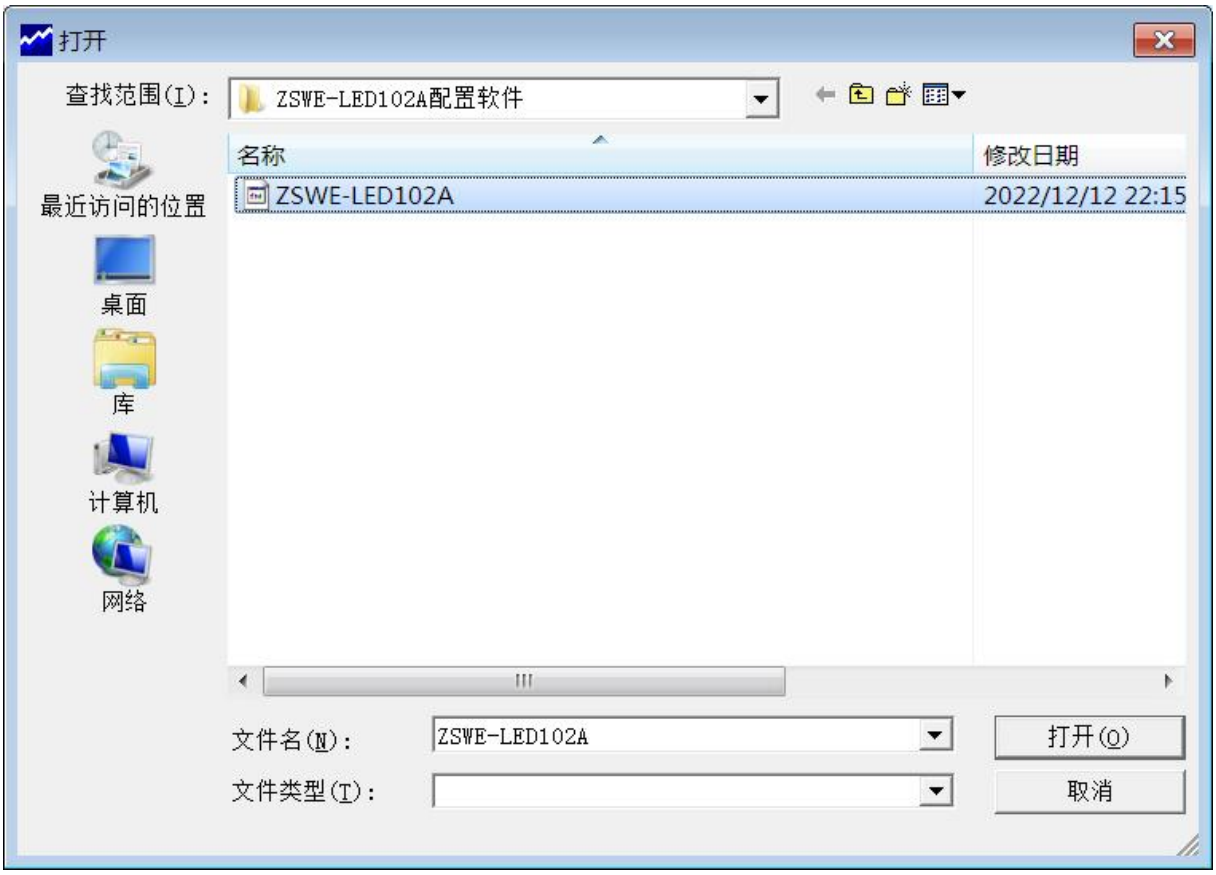
具体的配置软件操作方法，请参考后面的参数配置说明章节。

参数配置软件说明

1 配置软件说明

1.1 LED报警显示屏上电，SYS灯1秒一闪，说明设备工作正常，先等待10S，等模块启动。

1.2 下载参数配置软件，运行参数配置软件LEDCFG.EXE，第一次打开时参数区全是空的，需要点击文件操作-调入参数文件选择参数配置文件夹下的“ZSWE-LED102A.txt”，下一步就可以正常配置参数了。



说明：

LEDCFG 设置程序可以实现 LED报警显示屏参数的读取和设置，并且可以对屏 的工作状态进行测试。软件有“通信参数设置”、“透明传输测试”、“控制模式测试” 三个页面，点击某个页面即可进入相应功能界面，设置程序会自动向 LED报警显示屏发送各种工作模式切换命令，以便LED报警显示屏能够配合该软件进行相应的操作和测试。

设置程序通过串口或网线接口与 LED报警显示屏进行通信，从而完成各种操作。应该先在设置程序里面选择正确的串口波特率，以使计算机串口与 LED报警显示屏工作在相同的波特率，ZSWE-

LED102A LED报警显示屏出厂时的默认波特率为 9600,8N1。默认IP地址为192.168.1.61。

1.3 确定当前所用串口的串口号，修改串口号，并保持串口波特率一致，确认后点击“打开串口”。

串口打开成功后在软件的最下方边沿会显示串口打开成功。

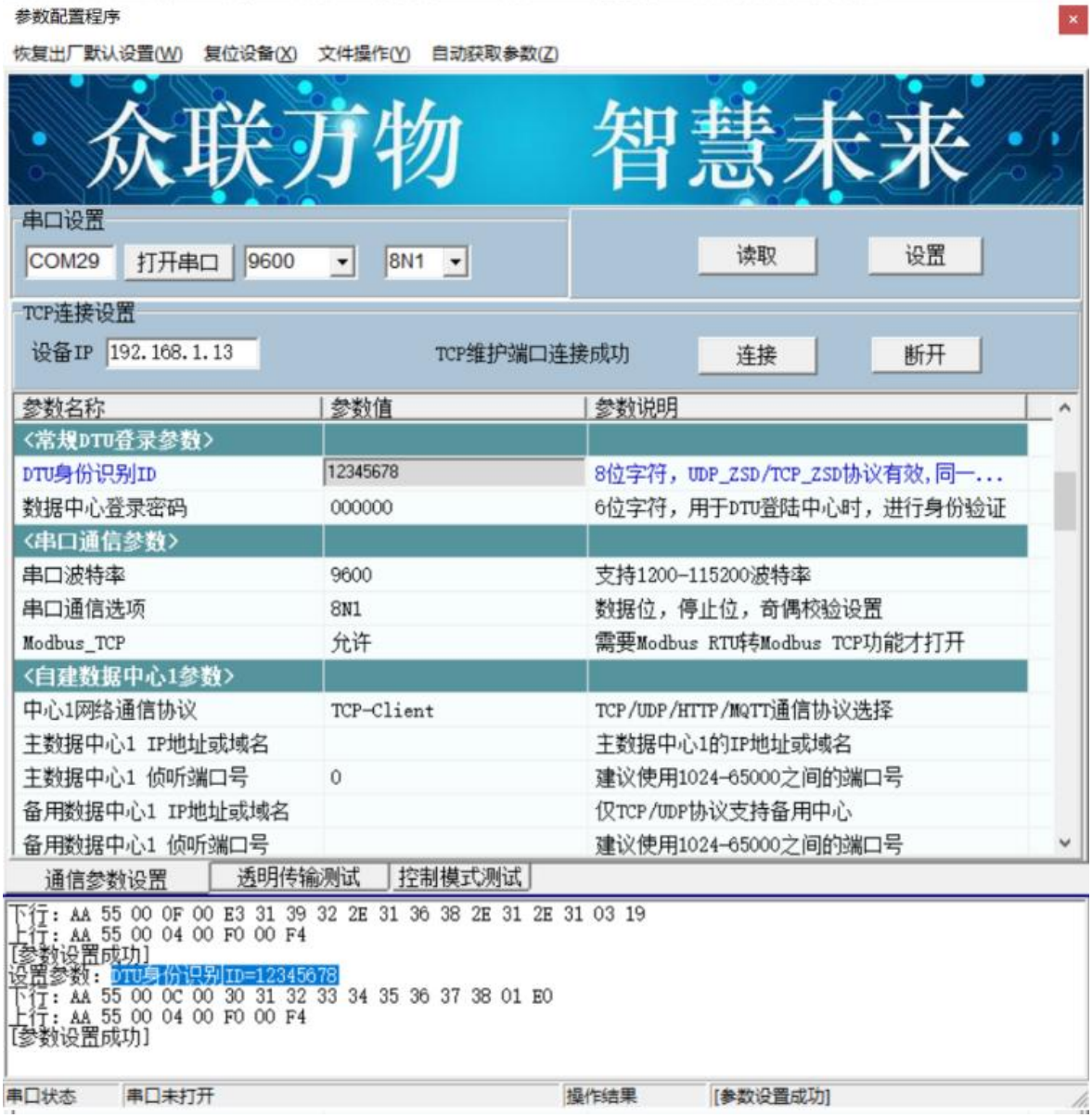
1.4 在“通信参数设置”页中，点击右上角的“读取”按钮，即可显示出当前的通信参数值，如下图：

（当设备联网成功时，也可以通过TCP连接方式进行参数设置），连接TCP维护端口成功后，可以进行参数读取和设置，操作与串口配置方式相同。

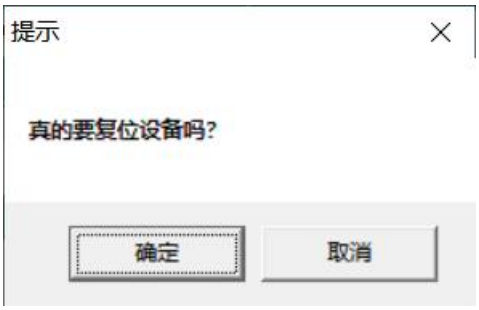
1.5 双击要修改的参数值，直接输入或修改相应的参数值，点击右上角的“设置”按钮即可完成 参数的设置。要使新参数生效，必须复位报警显示屏或者重新上电。

（注：设备出厂默认的IP地址为192.168.1.61，如果用户修改IP地址生效后，还需要继续通过TCP连接方式配置参数，则应该以新的IP地址去连接设备）

参数设置成功后，在下面的信息窗口中会有提示：



所有参数配置完以后点左上角复位按钮，然后点确认即可。



注：参数配置完以后需要复位设备，参数才会生效

脚本参数

LED报警显示屏通过脚本进行数据信息的显示和报警控制，以及通过脚本上报数据到服务器端。

脚本说明:

为实现设备操作的简易性和灵活性，我公司专门研发了一套控制脚本指令，通过脚本可以快速简洁地实现各种采集显示和报警控制功能。

设备提供2条脚本供设置，2条脚本是并行运行的。

脚本内置变量区，I1-I60为整数变量，F1-F60为浮点数变量。

基本格式

@cmd=value

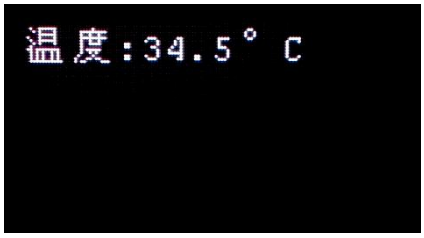
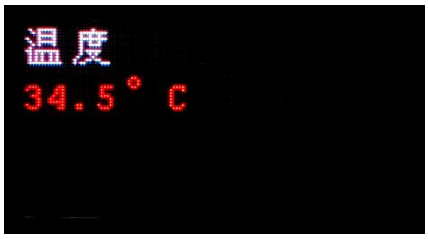
@：脚本头，每一条脚本指令都使用@开始。

cmd：指令，为 1 或多个字符、数字组成的字符串，不区分大小写，下面详解每一条脚本指令的含义。 value：指令动作，指定指令需要执行的值。

脚本设置在配置软件最底部的区域，双击内容框，可以弹出一个编辑窗口。


文本编辑框
@TEXT=F3R3G3B3,1,1,温度
@TEXT=F3R3G0B0,2,1,34.5°C
确定
内容长度：53字节

1. 显示控制脚本

指令	语法	说明
@TEXT 显示文字信息	@TEXT=显示属性，行，列，内容 显示属性：F?R?G?B?- F?对应显示是否闪烁，F0-不显示，F1-快闪，F2-慢闪，F3持续显示。 R?G?B? 控制显示颜色里面，红绿蓝三种颜色的比例。 ?对应为0-3的取值范围，控制每种颜色的四级显示亮度。可组合出64种颜色。 行：指定显示内容开始出现的行，1-4对应第1行到第4行 列：制定显示内容开始出现的列，1-16对应第1列到第16列； 内容：就是具体要显示的文本内容。可以直接是字符串，也可以是字符串混合格式化变量。 注：当内容超过一行时，会自动换行 F,R,G,B，以及行列值，均可由I1-I50变量进行代替，这样可以通过变量来实现显示效果的控制。	显示文本信息 例1： @TEXT=F3R3G3B3,1,1,温度:34.5℃ 效果：  例2： @TEXT=F3R3G3B3,1,1,温度 @TEXT=F3R3G0B0,2,1,34.5℃ 效果：  例3： @TEXT=F3R3G1B0,1,1,随着新发展格局加快构建，我国的绿色发展方式和生活方式将加快形成  例4： @SET=F1,15.1 @SET=F2,60.2 @SET=F3,1.23 @SET=F4,0.85 @TEXT=F3R3G0B0,1,1,温度:%.2f℃,F1 @TEXT=F3R3G3B3,2,1,湿度:%.2f%%,F2 @TEXT=F3R0G3B0,3,1,压力:%.2fmPa,F3

<table>
<tr><td></td><td></td><td>@TEXT=F3R1G3B3,4,1,液位:%.2f米,F4
 

F1-F4为浮点数变量，这里是直接设定变量值进行演示，变量可以和实际采集的传感器信号值进行关联赋值，具体方式请参考专门的脚本操作手册。</td></tr>
<tr><td>@TEXT24

显示24点阵字体的文字信息</td><td>@TEXT24=显示属性，字体，行，列，内容

显示属性：F?R?G?B?- F?对应显示是否闪烁，F0-不显示，F1-快闪，F2-慢闪，F3持续显示。

R?G?B? 控制显示颜色里面，红绿蓝三种颜色的比例。 ?对应为0-3的取值范围，控制每种颜色的四级显示亮度。可组合出64种颜色。

字体：ST-宋体，HT-黑体
行：指定显示内容开始出现的行，1-4对应第1行到第4行
列：制定显示内容开始出现的列，1-16对应第1列到第16列；
内容：就是具体要显示的文本内容。可以直接是字符串，也可以是字符串混合格式化变量。

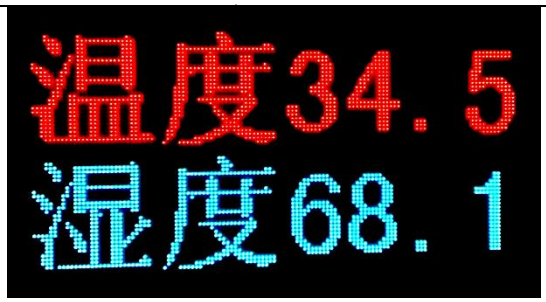
注：当内容超过一行时，会自动换行
F,R,G,B，以及行列值，均可由I1-I50变量进行代替，这样可以通过变量来实现显示效果的控制</td><td>显示文本信息
例1：
@TEXT24=F3R3G0B0,ST,1,1,@TEXT24=F3R3G3B3,ST,1,1,温度34.5
@TEXT24=F3R0G3B3,ST,3,1,@TEXT24=F3R3G3B3,ST,1,1,湿度68.1
效果：
 

例2：
@TEXT24=F3R3G0B0,HT,1,1,@TEXT24=F3R3G3B3,ST,1,1,温度34.5
@TEXT24=F3R0G3B3,HT,3,1,@TEXT24=F3R3G3B3,ST,1,1,湿度68.1
 
</td></tr>
<tr><td>@TEXT32

显示32点阵字体的文字信息</td><td>@TEXT32=显示属性，字体，行，列，内容

显示属性：F?R?G?B?- F?对应显示是否闪烁，F0-不显示，F1-快闪，F2-慢闪，F3持续显示。</td><td>显示文本信息
例1：
@TEXT32=F3R3G0B0,ST,1,1,温度34.5@D=0.01
@TEXT32=F3R0G3B3,ST,3,1,湿度68.1@D=0.01

效果：</td></tr>
</table>

	R?G?B? 控制显示颜色里面，红绿蓝三种颜色的比例。 ?对应为0-3的取值范围，控制每种颜色的四级显示亮度。可组合出64种颜色。 字体：ST-宋体 行：指定显示内容开始出现的行，1-4对应第1行到第4行 列：制定显示内容开始出现的列，1-16对应第1列到第16列； 内容：就是具体要显示的文本内容。可以直接是字符串，也可以是字符串混合格式化变量。 注：当内容超过一行时，会自动换行 F,R,G,B，以及行列值，均可由I1-I50变量进行代替，这样可以通过变量来实现显示效果的控制	
@TEXT48 显示48点阵字体的数字 注：48点阵字体，只支持数字0-9和小数点	@TEXT48=显示属性，字体，行，列，内容 显示属性：F?R?G?B?- F?对应显示是否闪烁，F0-不显示，F1-快闪，F2-慢闪，F3持续显示。 R?G?B? 控制显示颜色里面，红绿蓝三种颜色的比例。 ?对应为0-3的取值范围，控制每种颜色的四级显示亮度。可组合出64种颜色。 字体：ST-宋体 行：指定显示内容开始出现的行，1-4对应第1行到第4行 列：制定显示内容开始出现的列，1-16对应第1列到第16列； 内容：就是具体要显示的文本内容。可以直接是字符串，也可以是字符串混合格式化变量。 注：当内容超过一行时，会自动换行 F,R,G,B，以及行列值，均可由I1-I50变量进行代替，这样可以通过变量来实现显示效果的控制	显示文本信息 例： @TEXT=F3R3G3B0,1,1, 当前温度： @TEXT48=F3R3G0B0,ST,2,1,34.5@D=0.01 效果： 

@RECT 画矩形	@RECT=显示属性，X1,Y1,X2,Y2	例： @TEXT=F3R3G3B0,1,1, 当 前 温 度 ： @TEXT48=F3R3G0B0,ST,2,1,34.5@D=0.01 @RECT=F3R0G3B3,0,0,127,63 效果 
@LIINE 画线	@LINE=显示属性，X1,Y1,X2,Y2	例1： @TEXT=F3R3G3B0,1,1, 当 前 温 度 ： @TEXT48=F3R3G0B0,ST,2,1,34.5@D=0.01 @RECT=F3R0G3B3,0,0,127,63 @LINE=F3R3G3B3,0,16,127,16 效果： 
报警器控制	@DO1=1或者@DO1=0	例1： @DO1=1@D=1@DO1=0@D=10 每隔10秒报警器报警1秒钟。 例2： @IF=(F1>10)@DO1=1@ELSE@DO1=0@ENDIF 当F1变量大于10时，打开报警器，否则关闭报警器
信号采集	@C=采集报文@D=1 采集485总线上的传感器数据，LED屏内置的2路模拟量，也是挂接到485总线的，因此可通过信号采集脚本进行数据的采集	例： @C=010300000004V1@D=1 采集内部模拟量模块的2个通道的数据
变量赋值	@SET=变量命,字节取值地址	例：

		@C=010300000004V1@D=1 @SET=F1,FBB3 @SET=F2,FBB3 将第一路模拟量值赋值给F1,将第二路模拟量值赋值给F2
@LIGHT	亮度控制	亮度值从1-48 @LIGHT=24 设置为普通亮度 @LIGHT=48设置为最高亮度

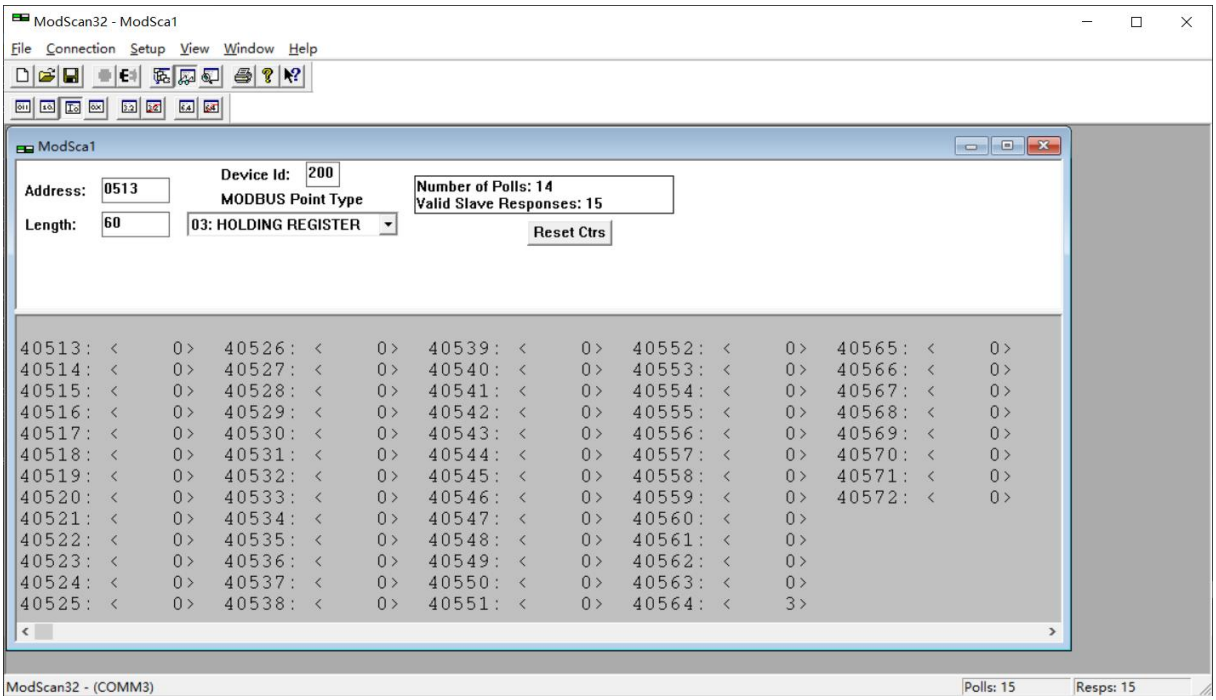
变量-MODBUS 映射区说明：

I1-I60这部分脚本内部的整数变量以及报警器的开关状态，除了可以通过脚本指令进行赋值和取值外，也进行了MODBUS寄存器映射，上位机可通过MODBUS RTU协议或者MODBUS TCP协议进行寄存器的读写，从而实现对脚本内部变量和报警器的操作。

1）LED屏默认MODBUS RTU地址：200

2）I0~I59 ： MODBUS寄存器:40513-40572 (内部地址：0x0200-0x023B)

3）报警器映射：MODBUS寄存器：40021 (内部地址：0x0014)



2. 逻辑控制及变量运算脚本

指令	名称	格式	含义
@C	向串口输出报文	@C=HEX @C=HEX+V1 @C-HEX+I1	定义执行命令，等效于中心下发数据，命令输入采用HEX格式，示例：@C=010203 向串口发送010203三个字节长度的包 当命令为MODBUS RTU 控制协议时，可以在命令最后使用V1 让 RTU 自动计算CRC校验并跟在命令的结尾，省去了用户自己计算校验的麻烦。此脚本指令定义的采集命令内容为可见的 HEX 字符。示例： Modbus 采集指令： 01 03 00 00 00 01 84 0A 脚本表示为： @C=010300000001V1 报文里面可以嵌入变量: 示例：@C=010203I1040506 如果I1=1，则会向串口发送01020301040506 如果I1=0，则会向串口发送01020300040506

指令	名称	格式	含义
@PRINT	向串口打印可见报文	@PRINT=STR @PRINT=Ix	定义执行命令，往串口发送打印内容，字符格式，功能类似于 @C，但不支持自动校验。字符内容不能出现@字符 PRINT： 执行命令，@PRINT定义的命令 RTU 功能和@C 相似，但@C 命令输入的是 HEX 格式命令，@PRINT输入的命令采用文本格式，适合于字符通讯协议的应用，免去用户转换成 HEX 格式命令的麻烦，也可以用于打印变量I1-I10的内容。 示例：

			1）@PRINT=HELLO ，向串口发送HELLO字符串 2）@PRINT=I1，向串口发送变量I1的值。

指令	名称	格式	含义
@PRTLN	向串口打印可见报文并换行	@PRTLN =STR	@PRINT输入的命令采用文本格式，适合于字符通讯协议的应用，免去用户转换成 HEX 格式命令的麻烦，也可以用于打印变量I1-I10的内容，字符内容不能出现@字符 示例： 3）@PRTLN =HELLO ，向串口发送HELLO字符串以及换行符 4）@PRTLN =I1，向串口发送变量I1的值以及换行符。

指令	名称	格式	含义
@D	延时控制	@D=X	延时控制，此脚本指令用于控制延时，X为数字的格式，单位为秒， 支持小数。RTU遇到此指令后等待相应的延时值再继续执行脚本。最小分辨率为0.001（1毫秒） 示例： @D=10 (等待 10 秒) @D=0.2 (等待0.2 秒) @D=60 (等待60秒)

指令	名称	格式	含义
@DO	继电器控制	@DOX=1,0,~ @DOX=DIX,!DIX @DOX=DOX,!DOX @DOX=(I1<5)	控制继电器动作 示例： @DO1=1 , 继电器1打开 @DO1=0 , 继电器1关闭 @DO1=~ , 继电器1状态翻转 @DO1=DI1 , 继电器1状态与DI1同步 @DO1=!DI1 , 继电器1状态与DI1相反 @DO2=DO1 , 继电器2状态与继电器1状态同步 @DO2=!DO1 , 继电器2状态与继电器1状态相反 @DO1=(I1<5) 当变量I1小于5时，继电器1导

			通，否则断开

指令	名称	格式	含义
@SET	SET	@SET =Ix,y	给把y赋值给变量Ix 示例： @SET=I1,1 I1赋值为1 @SET=I1,2 I1赋值为2 说明，变量为I1-I10，整型。

指令	名称	格式	含义
@INC	变量值增加	@INC=Ix,[y]	变量Ix的值增加1或者y 示例： @INC=I1 I1值加1 @INC=I1,10 I1值加10
指令	名称	格式	含义
@DEC	变量值减小	@DEC=Ix,[y]	变量Ix的值减小1或者y 示例： @DEC=I1 I1值减1 @DEC=I1,10 I1值减10
指令	名称	格式	含义
@MUL	变量值乘	@MUL =Ix,[y]	变量Ix的值乘以y 示例：@MUL =I1,2 I1=I1*2
指令	名称	格式	含义
@DIV	变量值除	@DIV=Ix,[y]	变量Ix的值除以y 示例：@DIV =I1,2

指令	名称	格式	含义
@IF	条件判断	@IF @ENDIF @IF@ELSE @ENDIF	根据条件，判断是否执行代码 条件包括： 值判断：1,0 IO当前状态判断 DI1H,DI1L;DI2H,DI2L,...DIxH,DIxL DO1H,DO1L;DO2H,DO2L,...DOxH,DOxL IO状态变化判断 DI1~,DI2~;...DIx~ 发生变化 !DI1~,!DI2~;...!DIx~ 为发生变化 数据接收判断 (RD=NULL) 串口未接收数据 (RD<>NULL) 串口接收到数据 (RD=.....) 串口接收到.....开始的报文 (CD=NULL) 中心未下发数据 (CD<>NULL) 中心下发数据 (CD=.....) 中心下发的数据以.....开始 变量判断 (I1==X)...(I10==X)或(I10==X) (I1<X)或(I1<=X)...(I10<X)或(I10<=X) (I1>X)或(I1>=X)...(I10>X)或(I10>=X) (I2<X)或(I2<=X)...(I10<X)或(I10<=X) (I2>X)或(I2>=X)...(I10>X)或(I10>=X) (I1<>X);(I2<>X)...(I10<>X) (I1==I2)或(I1=I2)...(Ix==Iy) (I1<I2)或(I1<=I2)...(Ix<Iy)或(Ix<=Iy)

			(I1>I2)或(I1>=I2)...(Ix>Iy)或(Ix>=Iy) (I1<>I2)...(Ix<>Iy) 示例： @IF=DI1H@DO1=1@ENDIF@D=0.1 当DI1导通时，DO1导通 @IF=DI1H@DO1=1@ELSE@DO1=0@ENDIF@D=0.1 当DI1导通时，DO1导通,否则DO1断开 （等效于@DO1=DI1） @IF=(I1<5)@DO1=1@ELSE@DO1=0@ENDIF@D=0.1 当变量I1小于5时，DO1导通，否则断开 @IF=START@DO1=0@DO2=1@ENDIF@D=0.1 当程序启动时，继电器1断开，继电器2导通 @IF=(RD=010101)@DO1=1@ENDIF@IF=(RD=00000)@DO1=0@ENDIF@D=0.1 当串口收到010101时，DO1导通，当串口收到000000时，DO1断开 @IF=(RD=NULL)@DO1=1@ENDIF@D=0.1 当串口未收到任何数据包，DO1导通 @IF=(RD<>NULL)@DO1=1@ENDIF@D=0.1 当串口收到任意数据包，DO1导通 @IF=DI1~@DO1=1@ENDIF@D=0.1 当DI1状态发生了变化时，DO1导通 （注：IF可以嵌套使用，最多10层） 例如： @IF=DI1H@IF=DI2H@DO1=1@ELSE@DO1=0@ENDIF@ELSE@DO1=0@ENDIF 当DI1和DI2都导通时，DO1导通,否则DO1断开

指令	名称	格式	含义
@FOR	条件判断	@FOR=(条件) @ENDFOR	根据条件，判断是否执行代码 条件包括： 1,0,START DI1H,DI1L;DI2H,DI2L,...DIxH,DIxL DO1H,DO1L;DO2H,DO2L,...DOxH,DOxL DI1~,DI2~;...DIx~ !DI1~,!DI2~;...!DIx~ (RD=NULL);(RD<>NULL);(RD=.....) (CD=NULL);(CD<>NULL);(CD=.....) (I1==X)...(I10==X)或(I10==X) (I1<X)或(I1<=X)...(I10<X)或(I10<=X) (I1>X)或(I1>=X)...(I10>X)或(I10>=X) (I2<X)或(I2<=X)...(I10<X)或(I10<=X) (I2>X)或(I2>=X)...(I10>X)或(I10>=X) (I1<>X);(I2<>X)...(I10<>X) (I1==I2)或(I1=I2)...(Ix==Iy) (I1<I2)或(I1<=I2)...(Ix<Iy)或(Ix<=Iy) (I1>I2)或(I1>=I2)...(Ix>Iy)或(Ix>=Iy) (I1<>I2)...(Ix<>Iy) 示例： @SET=I1,0@FOR=(I1<10)@INC=I1@DO1=~@D=0.1@ENDIFOR 循环执行10次： DO1状态翻转，然后

			延迟0.1秒（注：FOR 可以嵌套使用，最多10层。） 例如： @SET=I1,0@FOR=(I1<10)@INC=I1@SET=I2,0@FOR=(I2<10)@INC=I2@DO1=~@D=0.5@ENDFOR@ENDFOR 效果：DO1翻转，然后延迟0.5秒。 总共执行100次。
@WCOM	提取串口数据	@WCOM=1 @WCOM=0	@WCOM=1 表示脚本需要将串口收到的数据进行逻辑处理，不会将串口数据发给数据中心 @WCOM=0 表示脚本不需要将串口收到的数据进行逻辑处理，会将串口数据发给数据中心
@WDCD	提取中心数据	@WDCD=1 @WDCD=0	@WDCD=1 表示脚本需要将从中心收到的数据进行逻辑处理，不会将数据发到串口 @WDCD=0 表示脚本不需要将从中心收到的数据进行逻辑处理，会将数据发给串口
@S	数据上报控制	@S=1 @S=0	@S=1 表示将上行缓冲区的数据立即上报给数据中心 @S=0 表示清除上行缓冲区的数据

3. JSON处理脚本

说明：设备脚本内置O1-O10，10个Json对象，以下指令可以对Json对象进行操作，包括将Object对象的赋值，取值，Object与字符串之间转换等。

指令	名称	格式	含义
@JADD	给JSON对象的数组添加值	@JADD=数据类型，对象，值 数据类型：I-整数，F-浮点数，S-字符串，L-数组，O-嵌套对象 对象：指定对象内插入值的节点名称 值：需要插入的数值，可以直接是整数值、浮点数值或者字符串值； 也可以是变量名：I1-I10,F1-F10,S1-S10,O1-O10	给JSON对象的数组添加值 例如： @JADD=I,O1.list,1212 添加整数1212到O1.list队列 @JADD=I,O1.list,I1 添加变量I1的值到O1.list队列 @JADD=F,O1.list,123.4 添加浮点数到O1.list队列 @JADD=S,O1.list,hello 添加字符串到O1.list @JADD=L,O1.list,sublist 添加嵌套数组到O1.list @JADD=O,O1.list,O2 添加嵌套对象O2到O1.list
@JSET	设置JSON对象的节点值	@JSET=数据类型，对象，值 数据类型：I-整数，F-浮点数，S-字符串，O-嵌套对象 对象：指定对象内设置该值的节点名称 值：需要设置的值或变量	设置JSON对象的节点值 例如： JSET=I,O1.year,12 将O1.year设置为12 JSET=I,O1.year,I1 将O1.year设置为I1的值 JSET=F,O1.weight,23.5 将O1.weight设置为23.5 JSET=S,O1.name,zhongshan 将O1.name设置为zhongshan JSET=O,O1.subobj,O2 将O1.subobj设置为O2
@JGET	获取JSON对象的节点值	@JGET=数据类型，对象，变量 数据类型：I-整数，F-浮点数，S-字符串，O-嵌套对象	获取JSON对象的节点值 例如： JGET=I,O1.year,I1 将O1.year的值

		对象：指定对象内设置该值的节点名称 变量：需要保存值的变量	赋值给变量I1 JGET=F,O1.weight,F1 将O1.weight的值赋值给变量F1 JGET=S,O1.name,S1 将O1.name赋值给变量S1 JGET=O,O1.subobj,O2 将O1.subobj赋值给O2
@LJGET	从对象数组中获取JSON对象	@JGET=序号，对象数组名，对象 序号： 整数，代表对象在数组中的序号，以0位开始 对象数组名：对象数组的名称 对象：保存获取对象内容的对象名称O2-O10	获取JSON对象的节点值 例如： LGET=0,O1.values,O2 将O1.values数组的第一个对象获取并保存到O2
@JCLR	清除对象的值	@JCLR=对象	清除对象的值 例如： @JCLR=O1 清除O1的值 @JCLR=O2 清除O2的值 @JCLR=ROOT 清除ROOT对象的值(O1-O10都会被清除)
@JDEC	将字符串解码为对象	@JDEC=字符串来源 字符串缓冲区： COM-从串口接收到的数据 DSC-从数据中心收到的数据	将字符串解码为对象 例如： JDEC=COM,O1 把从串口收到的字符串转换为O1 JDEC=DSC,O1 把从数据中心收到的字符串转换为O1
@JSTR	将对象转为字符串上报给数据中心	@JSTR=对象	将对象转为字符串上报给数据中心 例如： JSTR=O1 将对象O1转为字符串上报给数据中心
@JHEX	将对象转为HEX字符串文，上报给数据中心	@JHEX=对象	将对象转为HEX字符串上报给数据中心 例如： JHEX=O1 将对象O1转为HEX字符串上报给数据中心
@JCS	将对象转为字符串发到串口	@JCS=对象	将对象转为字符串发送给串口 例如： JCS=O1 将对象O1转为字符串上报给串口

//光伏应用脚本解释：

脚本1：

@TEXT=F3R3G3B3,1,1，光伏实时出力 //第1行，显示标题

//显示三行数据

@TEXT=F3R3G3B3,2,1,幕墙光伏:%5.1fKW,F1

@TEXT=F3R3G3B3,3,1,车棚光伏:%5.1fKW,F2

@TEXT=F3R3G3B3,4,1,屋顶光伏:%5.1fKW,F3

@D=0.5 //等待0.5秒

第2条脚本：

@WDCD=1
@D=1
//发送POST请求
@H=504F5354202F676174657761792F6170692F76312F6D6765632F7274646174613A717
565727920485454502F312E310D0A436F6E74656E742D547970653A206170706C6963617
4696F6E2F782D7777772D666F726D2D75726C656E636F6465640D0A486F73743A203137
322E34302E302E3234393A38300D0A436F6E74656E742D4C656E6774683A203133300D0
A200D0A7B226B657973223A205B227075625F77696E645F67656E657261746F723A3133
3A70222C0D0A227075625F736F6C61725F67656E657261746F723A35323A70222C22707
5625F736F6C61725F67656E657261746F723A35333A70222C0D0A227075625F736F6C61
725F67656E657261746F723A35343A70225D2C2274656E616E74223A202232227D@S=1
@D=3

@IF=(CD<>NULL)
//json解码
@JDEC=DSC,O1
@D=0.1
//获取values队列中的4个JSON对象
@LGET=0,O1.values,O2@JGET=F,O2.value,F1
@LGET=1,O1.values,O2@JGET=F,O2.value,F2

@LGET=2,O1.values,O2@JGET=F,O2.value,F3
@LGET=3,O1.values,O2@JGET=F,O2.value,F4

//取F1-F4绝对值

@IF=(F1<0)@MUL=F1,-1@ENDIF
@IF=(F2<0)@MUL=F2,-1@ENDIF
@IF=(F3<0)@MUL=F3,-1@ENDIF
@IF=(F4<0)@MUL=F4,-1@ENDIF

//将F3加上F4的值（车棚1+车棚2）

@INC=F3,F4

@JCLR=ROOT
@ENDIF

//等待0.1秒

@D=0.1

POST http://172.40.0.249