

Zport 系列 IND 模块_显控 HMI 示例程序说明

第一步,如图 1 所示,设置串口通讯参数为:115200,8,N,1,建立 Zport 模块 ModbusRTU 通讯, 通讯周期时间设置为 100ms (通讯时间应大于宏指令脚本周期时间)。

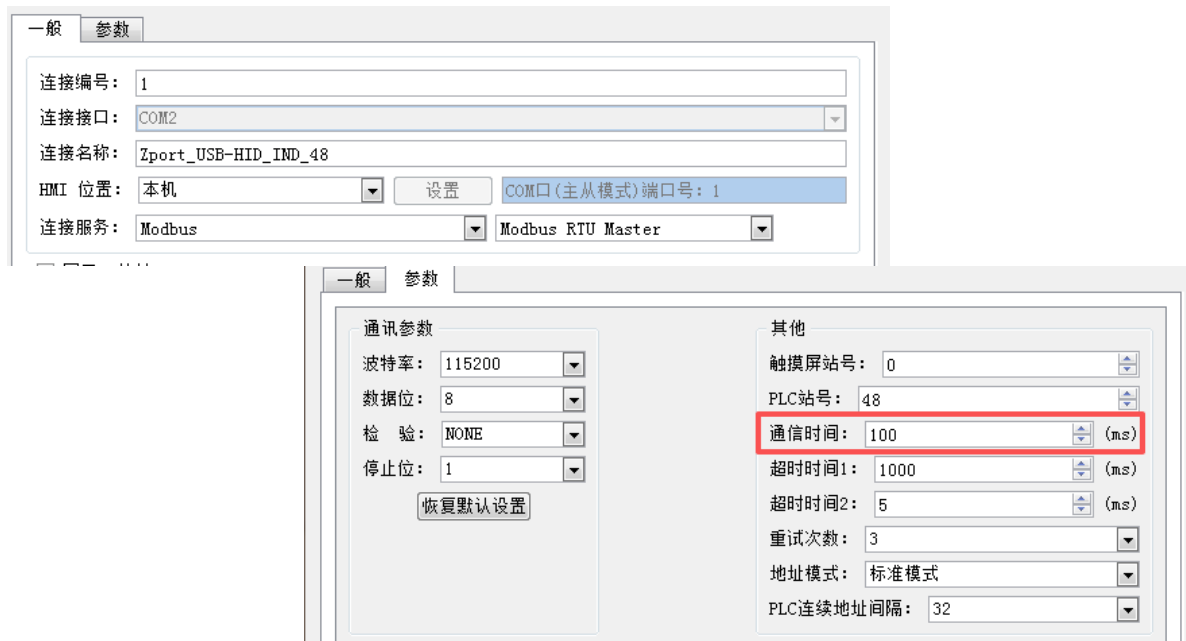


图 1

第二步,如图 2 所示,建立“字地址转发”,将 Modbus 地址为 3x1 (对应功能码 04, 起始地址 01) 的连续 30 个字寄存器的数据存入 HMI 内部寄存器地址 LW0..29。

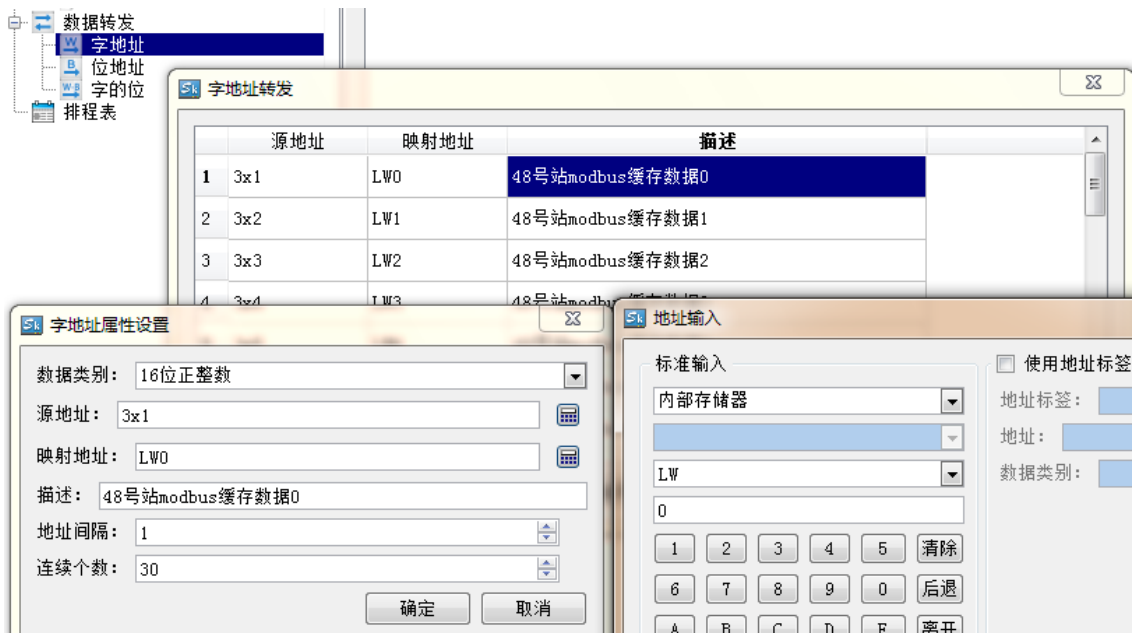


图 2

第三步,如图 3 所示,创建宏指令,设置全局变量 **Modbus 数据缓存变量“scanner48”**和导出字符串_内部存储变量名称 **“LW100”**,分别映射地址 LW0..29 和 LW100..114。设置全局变量 LB0 用于字符串比较测试,设置全局变量 LW99 作为“扫码计数器”,方便判断扫取相同码时扫码字符的新旧。

“**Modbus 数据缓存变量**”字节数与“**导出字符串_内部存储变量名称**”字节数为 2 倍关系。

“**导出字符串_内部存储变量名称**”字节数与第四步导出脚本之中,扫码字符串最大长度相同。



```

1 #include "MacroInit.h"
2 // #include "subFun0.c" //引用子程序示例
3
4 char ScannerStr[31]; //扫码字符串缓存: 30 个有效字符 + 1 个 \0
5 unsigned int ScannerBuffer[30]; //scanner48[] 副本, 每个 WORD 只取低 8 位
6 unsigned char i;
7 unsigned char j; //有效字符数
8 unsigned char k; //LW100[] 写入下标, 单位: WORD
9 unsigned char ch1;
10 unsigned char ch2;
11 unsigned char str_len; //扫码字符串最大长度, 对应 scanner48[30]
12 unsigned char word_len; //LW100 WORD 数量, word_len=(str_len+1)/2
13

```

变量名	数据类别	字节数 (B)	读写属性	寄存器	地址
1 scanner48	批量16位正整数	60	读/写	LW	0
2 LW100	批量16位正整数	30	读/写	LW	100
3 LB0	bool	1	读/写	LB	0
4 LW99	16位正整数	2	读/写	LW	99
5 新建...					

图 3

第四步,如图 4 所示,使用“Zport 系列 IND 模块显控 HMI—字符串脚本导出程序.html”脚本,生成脚本程序,参数设置中:

“**目标字符串变量名称**”做为字符串比较时使用。

“**Modbus 数据缓存变量名称**”对应宏指令“**scanner48**”全局变量;

“**导出字符串_内部存储变量名称**”对应宏指令“**LW100**”全局变量;

“**结尾字符串清除**”一般选择 0D/03 结束符,用于清除扫码自带 0x0D 回车字符和模块自带 0x03 结束字符;

“**扫码字符串最大长度**”的最小设置值,建议按照“项目最大扫码字符长度 + 3”进行设置。实际应用中通常取 10 的整数倍,便于配置和维护。需要注意的是,读取寄存器数量越多,单次通信的数据长度越大,收发时间也会相应增加,从而降低通信效率。因此,应根据实际扫码字符长度合理设置读取数量。

“**LW 字节组合格式**”按照显控默认设置格式一。



图 4

第五步，如图 5 所示，创建脚本程序，将第四步复制的程序代码粘贴、编译并保存后，添加至全局宏中，执行频率设置为 50ms，此时间应小于 modbus 通讯周期时间 100ms。

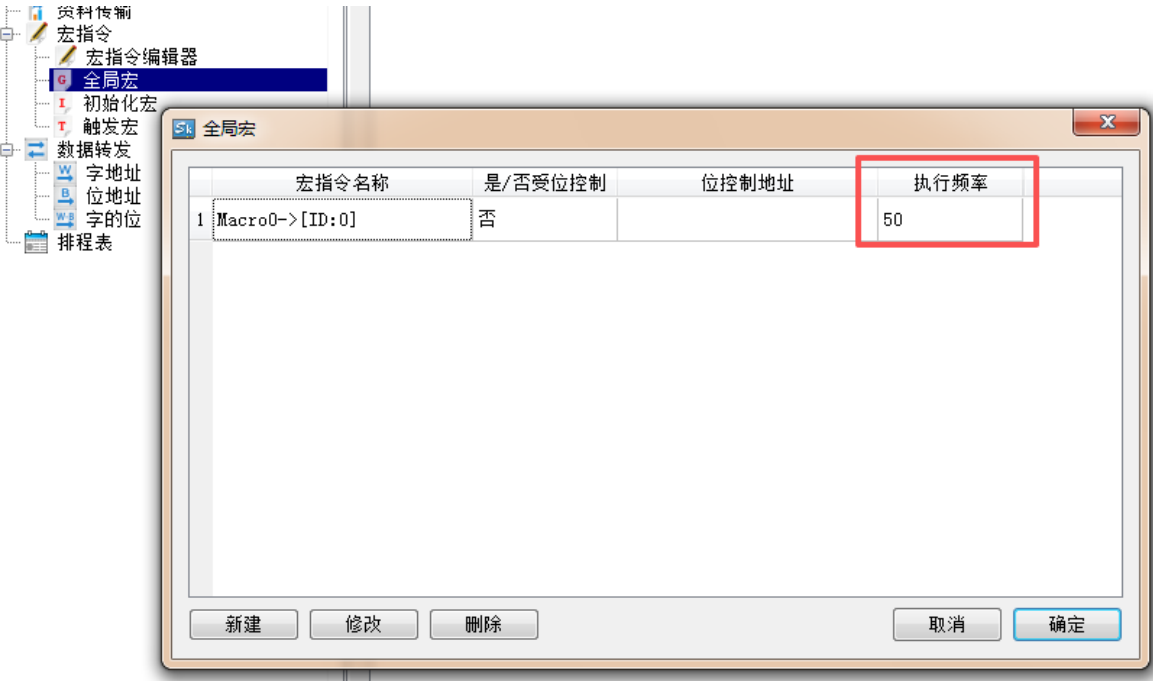


图 5

第六步，如图 6 所示，在宏指令尾部添加“用户程序”，用于字符串比较输出指示和扫码计数。比较测试字符串在宏指令头部 TestStr 数组中设置。

```

96      /* =====
97      以下添加用户程序
98      说明：此区域位于扫码解析和 LW 打包完成之后。
99      可在此添加字符串比较、数据校验、报警输出等项目逻辑。
100     下面为字符串比较示例，可按实际项目修改或删除。
101     ===== */
102     result = StringCompare(ScannerStr, TestStr);
103
104     if (result == 0)
105     {a
106         LED = 1;
107     }
108     else
109     {
110         LED = 0;
111     }
112
113     /* a比较相同LED点亮 */
114     LB0 = LED ;
115     /* 扫码计数器 */
116     LW99 = LW99 + 1;
117 }
118 }
119

```

编译信息输出 | 变量设定 | 全局变量设定

	变量名	数据类别	字节数(B)	读写属性	寄存器	地址	连接类型	
1	scanner48	批量16位正整数	60	读/写	LW	0	内部存储器:1	10
2	LW100	批量16位正整数	30	读/写	LW	100	内部存储器:1	10
3	LB0	bool	1	读/写	LB	0	内部存储器:0	10
4	LW99	16位正整数	2	读/写	LW	99	内部存储器:1	10
5	新建...							

图 6

第七步，如图 7 所示，在画面中，设置“ASCII 显示”和“数值显示”控件，并分别绑定“导出字符串_内部存储变量 LW100（长度 30）”和“扫码计数器 LW99”。字符串比较 LED 灯绑定 LB0。



图 7

第八步，编译下载后，进行测试。扫取界面“1234”二维码后，指示灯点亮，扫取其他码时熄灭。

