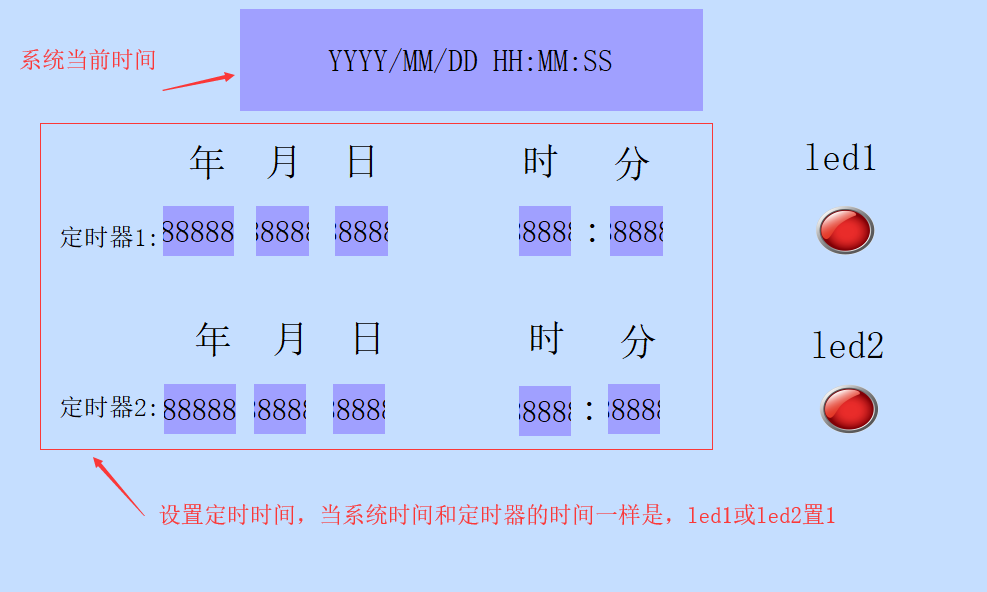
**宏指令案例**

**注：案例工程组态可在底部附件另存，也可在官网下载**

**官网下载地址：http://www.kinseal.com/download.asp?cd=11&cid=19&bg=xz**

**001：定时功能案例：**实现在触摸屏时间达到指定时间后，进行内部地址LB0的地址值反转。

1. 新建工程，做一个定时组态，如下图



（2）根据组态编辑定时器宏指令作定时判断，当年月日时分和当前时间一样时，指示灯显示为1，反之为0

宏指令编辑代码如下；

void Macro\_Entry( )

{

// TO DO

int year,month,day,hour,min;

//从内部特定地址中获取当前年、月、日、时、分、秒时间

year = BCD2BIN(LocalWord[3925]);

month = BCD2BIN(LocalWord[3924]);

day = BCD2BIN(LocalWord[3923]);

hour = BCD2BIN(LocalWord[3922]);

min = BCD2BIN(LocalWord[3921]);

//判断第一段定时时间是否到达

if(year == LocalWord[3000] && month == LocalWord[3001] && day == LocalWord[3002] &&\

hour == LocalWord[3003] &&min ==LocalWord[3004] )

{

//定时时间到，亮led1；

//注：led1是变量名，地址按需要设置

led1 = 1;

}

else

led1 = 0;

//判断第二段定时时间是否到达

if(year == LocalWord[3010] && month == LocalWord[3011] && day == LocalWord[3012] &&\

hour == LocalWord[3013] &&min ==LocalWord[3014] )

{

//定时时间到，亮led2；

//注：led2是变量名，地址按需要设置

led2 = 1;

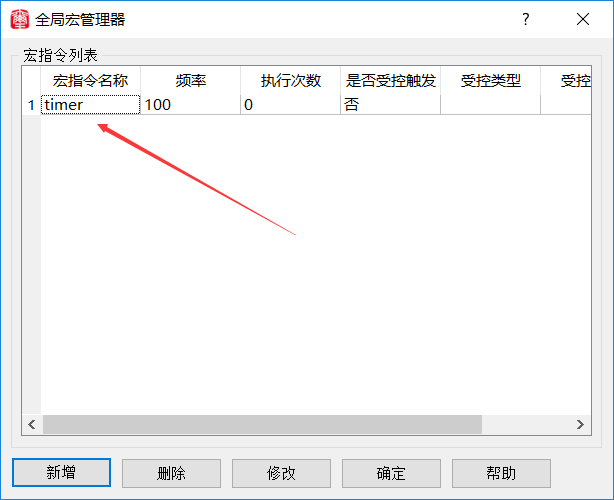
}

else

led2 = 0;

}

（3）代码编辑完成后点击右上角全部编译，下方提示编译成功时关闭窗口，然后然后在全局宏调用，如下图



（4）完成后下载组态到触摸屏即可看到定时效果

**002：计时功能案例：**实现触摸屏在达到累计时间后，进行报警提示。

1. 新建工程，做一个计时器，如下图



（2）根据组态编辑计时器宏指令作计时判断，设定总时间（单位：秒），当达到计时总时间时触发报警

宏指令编辑代码如下；

void Macro\_Entry( )

{

short NowHour,NowSec,NowWeek;

//读取RTC实时时间的秒数

NowSec = BCD2BIN(LocalWord[3920]);

//启动按钮按下

if((LocalBit[100]==1))

{

//如果RTC秒数发生变化，秒数递增记录一秒

if(NowSec != LocalWord[1])

{

//计算累计时间

LocalWord[2]++;

//给秒数标志位赋值

LocalWord[1] = NowSec;

LocalWord[1010]--;

//达到计时总时间

if((0==LocalWord[1010]))

{

//复位启动按钮

LocalBit[100]=0;

//在数位地址报警中设定了LB[10]作为报警地址

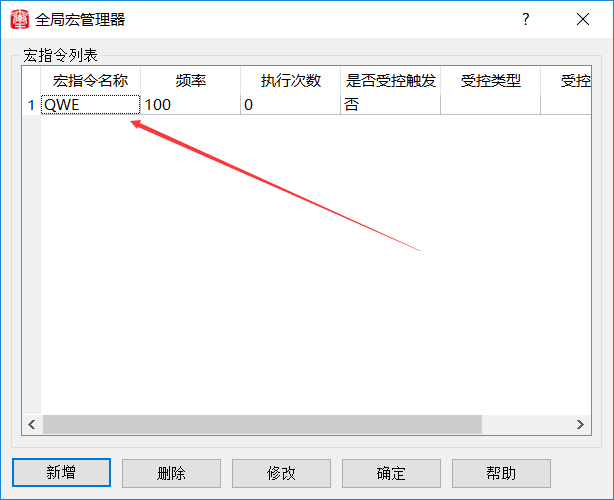
LocalBit[10]=1;

}

}

}

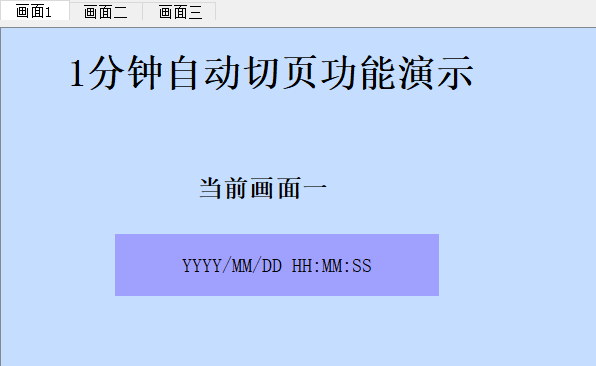
（3）代码编辑完成后点击右上角全部编译，下方提示编译成功时关闭窗口，然后在全局宏调用，如下图



（4）完成后下载组态到触摸屏即可看到定时效果

**003：画面跳转案例：**实现1分钟自动进行一次画面跳转。

1. 新建工程，建立多个画面，如下图



（2）根据组态编辑画面自动切换宏指令，实现1分钟自动进行一次画面跳转，

宏指令编辑代码如下；

void Macro\_Entry( )

{

short NowHour,NowSec,NowWeek;

//读取RTC实时时间的秒数

NowSec = BCD2BIN(LocalWord[3920]);

//如果RTC秒数发生变化，秒数递增记录一秒

if(NowSec != LocalWord[1])

{

//计算累计时间

LocalWord[2]++;

//给秒数标志位赋值

LocalWord[1] = NowSec;

//达到计时总时间,60秒

if((60==LocalWord[2]))

{

//计数标志清零

LocalWord[2] = 0;

//页面计数加1

LocalWord[3]++;

//调用函数库里面的函数，打开指定的画面

ScreenOpen(LocalWord[3]);

//当页面计数为2时，清零

if(2 == LocalWord[3])

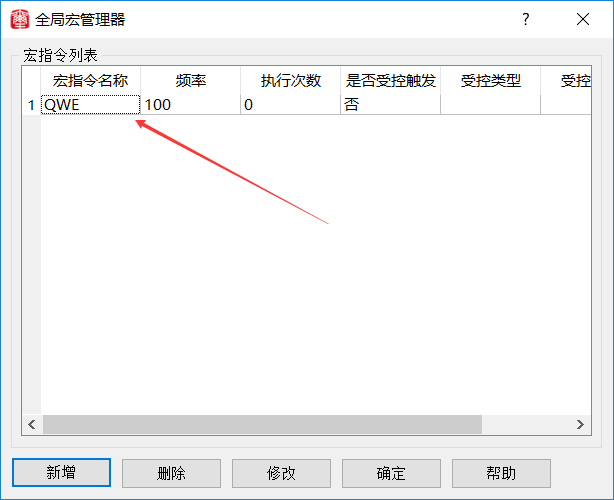
LocalWord[3] = 0;

}

}

}

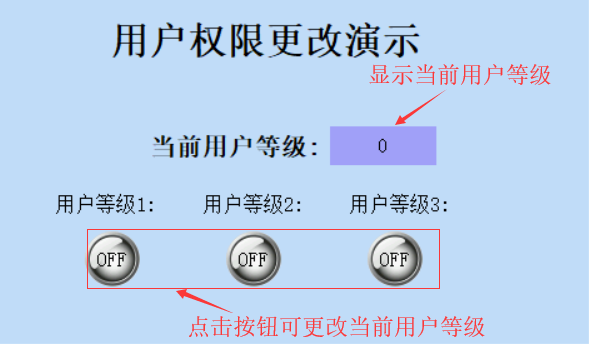
（3）代码编辑完成后点击右上角的全部编译，下方提示编译成功时关闭窗口，然后在全局宏调用



（4）完成后下载组态到触摸屏即可看到定时效果

**004：用户权限更改案例：**实现在不同条件下，触摸屏自动切换用户等级。

1. 新建工程，做一个组态，三个位按钮，用来使用宏控制用户等级，一个数值显示器，显示当前用户等级



（2）根据组态编辑宏指令实现不同条件下自动切换用户等级，

宏指令编辑代码如下；新建三个宏指令

//Change\_usr1（第一个宏指令）

void Macro\_Entry( )

{

//注意:在"系统设置->plc控制->控制当前权限等级"中，已经设定LW10用来控制用户等级

LocalWord[10] = 1;s

}

//Change\_usr2（第二个宏指令）

void Macro\_Entry( )

{

//注意:在"系统设置->plc控制->控制当前权限等级"中，已经设定LW10用来控制用户等级

LocalWord[10] = 2;

}

//Change\_usr3（第三个宏指令）

void Macro\_Entry( )

{

//注意:在"系统设置->plc控制->控制当前权限等级"中，已经设定LW10用来控制用户等级

LocalWord[10] = 3;

}

（3）代码编辑完成后点击右上角的全部编译，下方提示编译成功时关闭窗口，

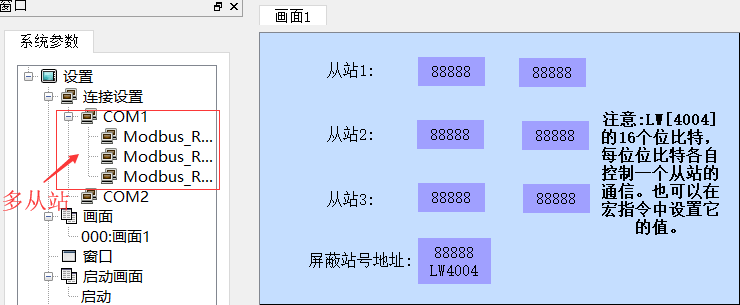


1. 每个按钮都各自使用一个宏，实现点击该按钮时更改用户等级



**005：从站通信屏蔽案例：**在一主多从用法时，可以选择屏蔽某些从站的通信。

1. 新建工程，做一个一组多从的工程



（2）做好组态后进行通讯，改变LW4004的值即可控制是否屏蔽某个从站

**006：微型打印机案例：**配合荣达系列微型打印机，打印指定内容。

1. 新建工程，做一个按钮，做打印按钮用



（2）编辑宏指令实现控制打印机打印指定的打印内容，

宏指令编辑代码如下；

void Macro\_Entry( )

{

char str[50];

char \*newptr,\*oldptr;

short source;//=LocalWord[3];

char result[4];

// char \*result=[4];

// short source =LocalWord[4] ;

// short source =LocalWord[3] ;

//把系统内部时间地址定义变量

//把PLC里的值传送给触摸屏系统内部地址

//LocalWord[1]=num8;

//LocalWord[2]=num1;

//LocalWord[3]=num2;

//LocalWord[4]=num3;

//LocalWord[5]=num4;

//LocalWord[6]=num5;

// LocalWord[7]=num6;

//LocalWord[8]=num7;

short NowYear ,NowMon,NowDay, NowHour,NowMin,NowSecond;

NowYear =BCD2BIN(LocalWord[3925]);

NowMon = BCD2BIN(LocalWord[3924]);

NowDay=BCD2BIN(LocalWord[3923]);

NowHour =BCD2BIN(LocalWord[3922]);

NowMin = BCD2BIN(LocalWord[3921]);

NowSecond=BCD2BIN(LocalWord[3920]);

LocalWord[9]=NowYear ;

LocalWord[10]=NowMon;

LocalWord[11]=NowDay;

LocalWord[12]=NowHour;

LocalWord[13]=NowMin;

LocalWord[14]=NowSecond;

if(LocalBit[100]==1)

{

newptr=(char\*)&LocalWord[1090];

sprintf(str,"\*\*\*\*\*");

oldptr=(char\*)&str;

memcpy(newptr,oldptr,5);

LocalWord[1095]=10;

LocalWord[1096]=10;

newptr=(char\*)&LocalWord[1090];

PUTCHARS(0,newptr,12);

//LocalWord[1102]=10;

newptr=(char\*)&LocalWord[1070];

sprintf(str,"额定转速");

oldptr=(char\*)&str;

memcpy(newptr,oldptr,8);

LocalWord[1078]=10;

newptr=(char\*)&LocalWord[1070];

PUTCHARS(0,newptr,26);

sprintf(str,"%d%02d%02d%02d%02d%02d ",NowYear,NowMon,NowDay,NowHour,NowMin,NowSecond);

newptr=(char\*)&LocalWord[1000];

oldptr=(char\*)&str;

memcpy(newptr,oldptr,14);

LocalWord[1015]=10;

newptr=(char\*)&LocalWord[1000];

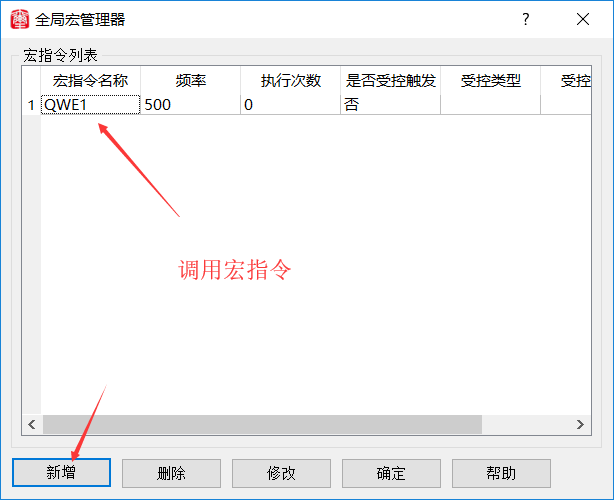
PUTCHARS(0,newptr,50);

LocalBit[100]=0;

}

}

（3）代码编辑完成后点击右上角的全部编译，下方提示编译成功，在全局宏调用



（4）下载组态到触摸屏，就可以控制打印机打印指定内容

**007：按钮宏案例：**实现一组按钮为互斥关系，即同一时刻，只能有一个按钮的值为1。

1. 新建工程，在画面上放置三个按钮



（2）编辑宏指令实现三个按钮互斥，只能有一个按钮的值能为1，

宏指令编辑代码如下；新建三个宏指令

//Change\_buttton1（第一个宏指令）

void Macro\_Entry( )

{

//第一个button按下，复位其他两个button

LocalBit[1] = 0;

LocalBit[2] = 0;

}

//Change\_buttton2（第二个宏指令）

void Macro\_Entry( )

{

//第二个button按下，复位其他两个button

LocalBit[0] = 0;

LocalBit[2] = 0;

}

//Change\_buttton3（第三个宏指令）

void Macro\_Entry( )

{

//第三个button按下，复位其他两个button

LocalBit[0] = 0;

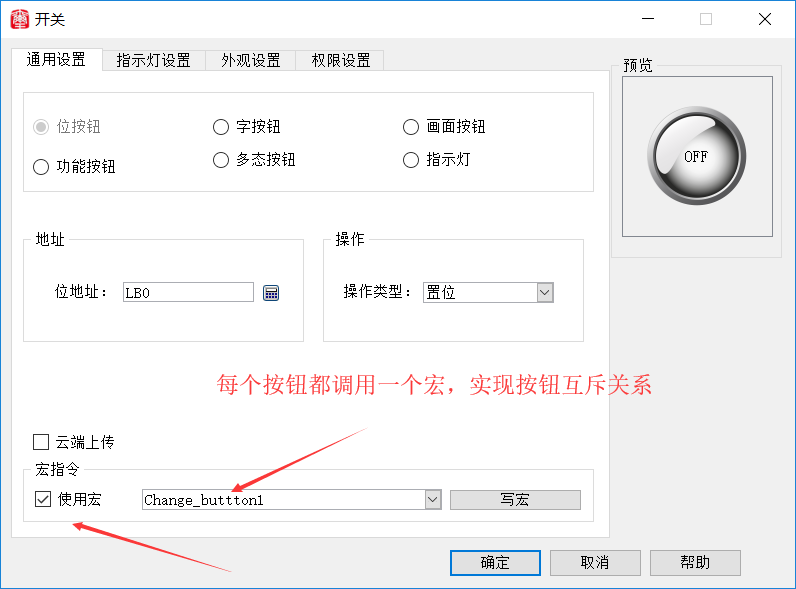
LocalBit[1] = 0;

}

（3）代码编辑完成后点击右上角的全部编译，下方提示编译成功时关闭窗口，



（4）给三个按钮都各自使用一个宏



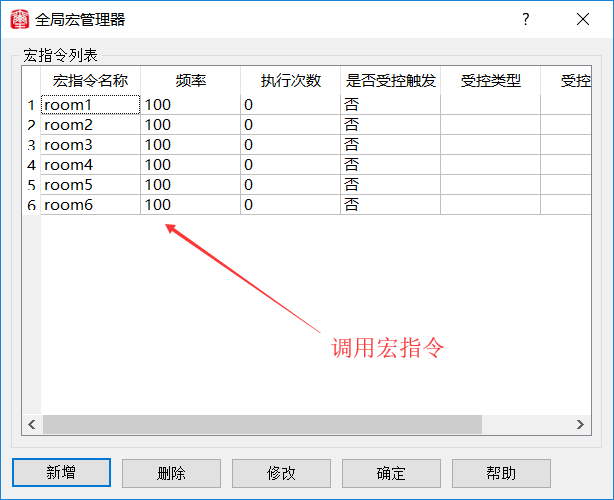
008、**二控一案例：**一个上位机、一个触摸屏与一个控制器，上位机与触摸屏同时可以控制控制器

1. 新建工程，做一个一组多从的组态，



1. 根据需求编辑宏指令，宏指令见附件组态（08、二控一案例）

（3）代码编辑完成后点击右上角的全部编译，下方提示编译成功，然后在全局宏调用



（4）最后把组态下载到触摸屏即可查看功能效果

附件：



